NON-CONFIDENTIAL

2013-1092, -1093, -1095, -1097, -1098, -1099, -1100, -1101, -1103

UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

MORGAN STANLEY, THE GOLDMAN SACHS GROUP, INC.,
J.P. MORGAN CHASE & CO., MORGAN STANLEY & CO., INC.,
GOLDMAN SACHS & CO., GOLDMAN SACHS EXECUTION &
CLEARING, LP, J.P. MORGAN SECURITIES, INC.,
and J.P. MORGAN CLEARING CORP.
(formerly known as Bear, Stearns Securities Corp.),
Defendants-Appellees,
and

CREDIT SUISSE HOLDINGS (USA), INC. and
CREDIT SUISSE SECURITIES (USA), LLC,
Defendants-Appellees,
and

HSBC BANK USA, N.A. and
HSBC SECURITIES (USA), INC.,
Defendants-Appellees,
and

BNY CONVERGEX GROUP, LLC and
BNY CONVERGEX EXECUTION SOLUTIONS, LLC,
Defendants.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

CME GROUP, INC., BOARD OF TRADE OF THE CITY OF CHICAGO, INC.,
and NEW YORK MERCANTILE EXCHANGE, INC. (agent of Nymex),
Defendants-Appellees,

and

BATS TRADING, INC. (also known as BATS Exchange, Inc.),
Defendant-Appellee,
and

INTERNATIONAL SECURITIES EXCHANGE,
Defendant-Appellee,
and

NASDAQ OMX GROUP, INC. and NASDAQ OMX PHLX, INC.,
Defendants-Appellees,
and

NYSE EURONEXT, OPTIONS PRICE REPORTING AUTHORITY,
NYSE ARCA, INC., NYSE MKT, LLC (formerly known as NYSE Amex, LLC),
and SECURITIES INDUSTRY AUTOMATION CORPORATION,
Defendants-Appellees.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

THOMSON REUTERS CORPORATION,
Defendant-Appellee,
and

BLOOMBERG, L.P.,
Defendant-Appellee,
and

FACTSET RESEARCH SYSTEMS, INC.,
Defendant-Appellee,
and

INTERACTIVE DATA CORPORATION,
Defendant-Appellee.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

CME GROUP, INC., BOARD OF TRADE OF THE CITY OF CHICAGO, INC., and NEW YORK MERCANTILE EXCHANGE, INC. (agent of Nymex),
Defendants-Appellees,
and

BATS TRADING, INC. (also known as BATS Exchange, Inc.),
Defendant-Appellee,
and

INTERNATIONAL SECURITIES EXCHANGE,
Defendant-Appellee,
and

NASDAQ OMX GROUP, INC. and NASDAQ OMX PHLX, INC.,
Defendants-Appellees,
and

NYSE EURONEXT, OPTIONS PRICE REPORTING AUTHORITY, NYSE ARCA, INC., NYSE MKT, LLC (formerly known as NYSE Amex, LLC), and SECURITIES INDUSTRY AUTOMATION CORPORATION,
Defendants-Appellees.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

THOMSON REUTERS CORPORATION,
Defendant-Appellee,
and

BLOOMBERG, L.P.,
Defendant-Appellee,
and

FACTSET RESEARCH SYSTEMS, INC.,
Defendant-Appellee,
and

INTERACTIVE DATA CORPORATION,
Defendant-Appellee.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

MORGAN STANLEY, THE GOLDMAN SACHS GROUP, INC.,
J.P. MORGAN CHASE & CO., MORGAN STANLEY & CO., INC.,
GOLDMAN SACHS & CO., GOLDMAN SACHS EXECUTION & CLEARING,
LP, J.P. MORGAN SECURITIES, INC., and J.P. MORGAN CLEARING CORP.
(formerly known as Bear, Stearns Securities Corp.),
Defendants-Appellees,
and

CREDIT SUISSE HOLDINGS (USA), INC. and
CREDIT SUISSE SECURITIES (USA), LLC,
Defendants-Appellees,
and

HSBC BANK USA, N.A. and
HSBC SECURITIES (USA), INC.,
Defendants-Appellees,
and

BNY CONVERGEX GROUP, LLC,
BNY CONVERGEX EXECUTION SOLUTIONS, LLC,
THE BANK OF NEW YORK MELLON CORPORATION,
BANK OF AMERICA CORPORATION,
BANC OF AMERICA SECURITIES, LLC,
MERRILL LYNCH & CO., INC., and
MERRILL LYNCH, PIERCE, FENNER & SMITH, INC.,
Defendants.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

CME GROUP, INC., BOARD OF TRADE OF THE CITY OF CHICAGO, INC., and NEW YORK MERCANTILE EXCHANGE, INC. (agent of Nymex),
Defendants-Appellees,
and

BATS TRADING, INC. (also known as BATS Exchange, Inc.),
Defendant-Appellee,
and

INTERNATIONAL SECURITIES EXCHANGE,
Defendant-Appellee,
and

NASDAQ OMX GROUP, INC. and NASDAQ OMX PHLX, INC.,
Defendants-Appellees,
and

NYSE EURONEXT, OPTIONS PRICE REPORTING AUTHORITY, NYSE ARCA, INC., NYSE MKT, LLC (formerly known as NYSE Amex, LLC), and SECURITIES INDUSTRY AUTOMATION CORPORATION,
Defendants-Appellees.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

THOMSON REUTERS CORPORATION,
Defendant-Appellee,
and

BLOOMBERG, L.P.,
Defendant-Appellee,
and

FACTSET RESEARCH SYSTEMS, INC.,
Defendant-Appellee,
and

INTERACTIVE DATA CORPORATION,
Defendant-Appellee.

REALTIME DATA, LLC (doing business as IXO),
Plaintiff-Appellant,
v.

MORGAN STANLEY, THE GOLDMAN SACHS GROUP, INC.,
J.P. MORGAN CHASE & CO., MORGAN STANLEY & CO., INC.,
GOLDMAN SACHS & CO., GOLDMAN SACHS EXECUTION & CLEARING,
LP, J.P. MORGAN SECURITIES, INC., and J.P. MORGAN CLEARING CORP.
(formerly known as Bear, Stearns Securities Corp.),
Defendants-Appellees,
and

CREDIT SUISSE HOLDINGS (USA), INC. and
CREDIT SUISSE SECURITIES (USA), LLC,
Defendants-Appellees,
and

HSBC BANK USA, N.A. and
HSBC SECURITIES (USA), INC.,
Defendants-Appellees,
and

BNY CONVERGEX GROUP, LLC,
BNY CONVERGEX EXECUTION SOLUTIONS, LLC,
THE BANK OF NEW YORK MELLON CORPORATION,
BANK OF AMERICA CORPORATION,
BANC OF AMERICA SECURITIES, LLC,
MERRILL LYNCH & CO., INC., and
MERRILL LYNCH, PIERCE, FENNER & SMITH, INC.,
Defendants.

Appeals from the United States District Court for the Southern District of New York in case nos. 11-CV-6696 through 11-CV-6704, Judge Katherine B. Forrest.

NON-CONFIDENTIAL BRIEF FOR PLAINTIFF-APPELLANT REALTIME DATA, LLC (doing business as IXO)

Robert A. Cote
Brett E. Cooper
Daniel J. Melman
Laura A. Handley
Lauren L. Fornarotto
MCKOOL SMITH P.C.
One Bryant Park, 47th Floor
New York, New York 10036
(212) 402-9400

Dirk D. Thomas
MCKOOL SMITH P.C.
1999 K Street, Suite 600
Washington, DC 20006
(202) 370-8302

Joel L. Thollander
MCKOOL SMITH P.C.
300 W. 6th Street, Suite 1700
Austin, TX 78701
(512) 692-8735

Attorneys for Plaintiff-Appellant Realtime Data, LLC (doing business as IXO)

March 6, 2013

CERTIFICATE OF INTEREST

Counsel for Plaintiff-Appellant Realtime Data, LLC (doing business as IXO) certifies the following:

1. The full name of every party represented by me is:

   REALTIME DATA, LLC (doing business as IXO).

2. The name of the real party in interest represented by me is:

   None.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party represented by me is:

   None.

4. The names of all law firms and the partners or associates that appeared for the party represented by me in the trial court or are expected to appear in this Court are:

   McKool Smith P.C.; Dirk Thomas, Robert Cote, Brett Cooper, Daniel Melman, Laura Handley, Lauren Fornarotto, John Briody, Kevin Schubert, Jonathan Yim, Warren Lipschitz, Pierre Hubert, Holly Engelmann, Samuel Baxter, Thomas Guy Fasone, Joel L. Thollander.

TABLE OF CONTENTS

CERTIFICATE OF INTEREST i

TABLE OF AUTHORITIES v

STATEMENT OF RELATED CASES viii

I. STATEMENT OF JURISDICTION 1

II. STATEMENT OF THE ISSUES 1

III. STATEMENT OF THE CASE 3

IV. STATEMENT OF FACTS 7

A. Realtime Invents Transformative Data Compression Technologies for the Financial Industry. 7

B. Financial Industry Players Find Realtime's Technology Provides "Impressive" Results with "Incredible Speed," but Elect Not to Take Licenses. 11

C. The FAST Standard Incorporates Realtime's Technology. 12

D. Realtime Brings Suit; the District Court Grants Summary Judgment on Every Asserted Claim. 15

V. SUMMARY OF ARGUMENT 19

VI. STANDARDS OF REVIEW 22

VII. ARGUMENT 23

A. The District Court Erred in Granting Summary Judgment of Noninfringement on the Basis of Three Misconstrued Terms That, Collectively, Cover Every Asserted Claim. 23

1. The court misconstrued "descriptor indicates." 23

2. The court misconstrued "data field/block type." 25

3. The court misconstrued "data stream." 30

4. The court erred in granting summary judgment of noninfringement on the basis of these misconstrued terms. 34

B. The District Court Further Erred in Granting Summary Judgment of Noninfringement on the Basis of These Terms Because, Even as Construed, Disputes of Material Fact Remain. 35

1. There is a genuine dispute regarding whether the accused products meet the "descriptor indicates" limitation. 36

2. There is a genuine dispute regarding whether the accused products meet the "data field/block type" limitation. 39

3. There is a genuine dispute regarding whether the accused products meet the "data stream" limitation. 43

C. The District Court Erred in Granting Summary Judgment of Noninfringement on Claims 95, 97, 108, and 112 of the '651 Patent By Applying an Agreed Construction to the Claims in a Manner That Excludes the Preferred Embodiment. 48

1. The agreed construction did not import an encoding requirement into the asserted decoding claims. 49

2. This Court should consider providing a clarifying construction for trial on remand. 52

D. The District Court Erred in Granting Summary Judgment of Invalidity on Claims 1, 4, 6, 7, and 12 of the ’651 Patent and Claims 1, 7, 8, and 13 of the ’747 Patent. 53

1. These claims comply with the definiteness requirement. 54

2. These claims comply with the written description requirement. 57

E. The District Court Abused Its Discretion in Precluding Realtime From Asserting Infringement Under the Doctrine of Equivalents. 59

VIII. CONCLUSION AND RELIEF REQUESTED 62

CONFIDENTIAL MATERIAL OMITTED

The material omitted on page 12 describes a Defendant-Appellee’s internal communications designated by that Defendant-Appellee as confidential pursuant to a Protective Order entered in the district court. The material omitted on pages 40 and 44 describes Defendants-Appellees’ accused instrumentalities based on source code designed by Defendants-Appellees as confidential pursuant to the Protective Order.

TABLE OF AUTHORITIES

Page(s)

CASES

Abbott Labs. v. Church & Dwight Co., No. 07-C-3428,
2008 U.S. Dist. LEXIS 103635 (N.D. Ill. Dec. 22, 2008)....................................32

Absolute Software, Inc. v. Stealth Signal, Inc.,
659 F.3d 1121 (Fed. Cir. 2011) 22, 38, 42, 45

Anderson v. Liberty Lobby, Inc.,
477 U.S. 242 (1986)...22

Bancorp Servs., L.L.C. v. Hartford Life Ins. Co.,
359 F.3d 1367 (Fed. Cir. 2004) 54, 57

Burke, Inc. v. Bruno Indep. Living Aids, Inc.,
183 F.3d 1334 (Fed. Cir. 1999) 34

Cybor Corp v. FAS Techs., Inc.,
138 F.3d 1448 (Fed. Cir. 1998) (en banc) 22, 23, 29, 60

Decisioning.com, Inc. v. Federated Dep't Stores, Inc.,
527 F.3d 1300 (Fed. Cir. 2008) 52

Energy Transp. Group, Inc. v. William Demant Holding A/S,
697 F.3d 1342 (Fed. Cir. 2012) 25, 27

Exxon Research & Eng'g Co. v. United States,
265 F.3d 1371 (Fed. Cir. 2001) 54

ICU Med., Inc. v. Alaris Med. Sys.,
558 F.3d 1368 (Fed. Cir. 2009) 22, 23

Int'l Visual Corp. v. Crown Metal Mfg. Co.,
991 F.2d 768 (Fed. Cir. 1993) 34

Interactive Gift Express, Inc. v. Compuserve Inc.,
256 F.3d 1323 (Fed. Cir. 2001) 52

Key Pharms. v. Hercon Labs. Corp.,
161 F.3d 709 (Fed. Cir. 1998) 29, 32

Laryngeal Mask Co. Ltd. v. Ambu A/S,
618 F.3d 1367 (Fed. Cir. 2010) 59

Marine Polymer Techs., Inc. v. HemCon, Inc.,
659 F.3d 1084 (Fed. Cir. 2011) 32

Martek Biosciences Corp. v. Nutrinova, Inc.,
579 F.3d 1363 (Fed. Cir. 2009) 24

Meyer Intellectual Props. Ltd. v. Bodum Inc.,
690 F.3d 1354 (Fed. Cir. 2012) 53

Mintz v. Dietz & Watson, Inc.,
679 F.3d 1372 (Fed. Cir. 2012) 23

Omega Eng'g Inc. v. Raytek Corp.,
334 F.3d 1314 (Fed. Cir. 2003) 33

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) 24, 25, 27, 29

Praxair, Inc. v. ATMI, Inc.,
543 F.3d 1306 (Fed. Cir. 2008) 22

Realtime Data, LLC v. MetroPCS Texas, LLC, No. 6:10-CV-493,
2012 U.S. Dist. LEXIS 141987 (E.D. Tex. Oct. 1, 2012)..........................33, 34

Realtime Data, LLC v. Morgan Stanley,
875 F. Supp. 2d 276 (S.D.N.Y. 2012) 5

Realtime Data, LLC v. Morgan Stanley, No. 11-CV-6696-KBF,
2012 U.S. Dist. LEXIS 89561 (S.D.N.Y. June 27, 2012) 5

Realtime Data, LLC v. Morgan Stanley, No. 11-CV-6696-KBF,
2012 U.S. Dist. LEXIS 109954 (S.D.N.Y. Aug. 2, 2012)....................................6

Realtime Data, LLC v. Morgan Stanley, No. 11-CV-6696-KBF,
2012 U.S. Dist. LEXIS 136511 (S.D.N.Y. Sept. 24, 2012) 6

Realtime Data, LLC v. Stanley, No. 11-CV-6696-KBF,
2012 U.S. Dist. LEXIS 161436 (S.D.N.Y. Nov. 9, 2012)....................................6

Schindler Elevator Corp. v. Otis Elevator Co.,
593 F.3d 1275 (Fed. Cir. 2010) 22

SiRF Tech., Inc. v. ITC,
601 F.3d 1319 (Fed. Cir. 2010) 50

Sri Int'l v. Matsushita Elec. Corp.,
775 F.2d 1107 (Fed. Cir. 1985) 30

Sundance, Inc. v. Demonte Fabricating Ltd.,
550 F.3d 1356 (Fed. Cir. 2008) 58

Ultimax Cement Mfg. Corp. v. CTS Cement Mfg. Corp.,
587 F.3d 1339 (Fed. Cir. 2009) 54, 56

Uniloc USA, Inc. v. Microsoft Corp.,
632 F.3d 1292 (Fed. Cir. 2011) 50, 51

Vita-Mix Corp. v. Basic Holding, Inc.,
581 F.3d 1317 (Fed. Cir. 2009) 28, 61

Vitronics Corp. v. Conceptronic, Inc.,
90 F.3d 1576 (Fed. Cir. 1996) 47, 52

STATUTES

28 U.S.C. § 1295(a) 1

28 U.S.C. § 1331 1

28 U.S.C. § 1338(a) 1

28 U.S.C. § 2107(a) 1

35 U.S.C. § 112 ....5, 21

OTHER AUTHORITIES

FED. R. APP. P. 4(a) 1

FED. R. CIV. P. 54(b) 6

FED. R. CIV. P. 56(a) 22

STATEMENT OF RELATED CASES

Pursuant to FED. CIR. R. 47.5, Plaintiff-Appellant Realtime Data, LLC (doing business as IXO) ("Realtime") provides as follows:

(a) There have been no other previous appeals in this case;

(b) The following related cases are pending in this Court and will directly affect or be directly affected by this Court's decision in the pending appeal:

REALTIME DATA, LLC V MORGAN STANLEY, 2013-1092;[1]

REALTIME DATA, LLC V CME GROUP, INC., 2013-1093;[2]

REALTIME DATA, LLC V THOMSON REUTERS CORP., 2013-1095;[3]

---

[1] The Defendants-Appellees in the Morgan Stanley cases are: Morgan Stanley and Morgan Stanley & Co., Inc. (collectively, "Morgan Stanley"); The Goldman Sachs Group, Inc., Goldman Sachs & Co., and Goldman Sachs Execution & Clearing, LP (collectively, "Goldman Sachs"); J.P. Morgan Chase & Co., J.P. Morgan Securities, Inc., and J.P. Morgan Clearing Corp. (formerly known as Bear, Stearns Securities Corp.) (collectively, "J.P. Morgan"); Credit Suisse Holdings (USA), Inc. and Credit Suisse Securities (USA), LLC (collectively, "Credit Suisse"); and HSBC Bank USA, N.A. and HSBC Securities (USA), Inc. (collectively, "HSBC").

[2] The Defendants-Appellees in the CME Group, Inc. cases are: CME Group Inc., Board of Trade of the City of Chicago, Inc., and New York Mercantile Exchange, Inc. (collectively, "CME"); BATS Trading, Inc. (also known as BATS Exchange, Inc.) ("BATS"); International Securities Exchange ("ISE"); NASDAQ OMX Group, Inc. and NASDAQ OMX PHLX, Inc. (collectively, "NASDAQ"); NYSE Euronext, NYSE Arca, Inc., NYSE MKT, LLC (formerly known as NYSE Amex, LLC), and Securities Industry Automation Corporation (collectively, "NYSE"); and Options Price Reporting Authority ("OPRA").

[3] The Defendants-Appellees in the Thomson Reuters Corp. cases are: Thomson Reuters Corp. ("Thomson Reuters"); Bloomberg L.P. ("Bloomberg"); FactSet Research Systems, Inc. ("FactSet"); and Interactive Data Corp. ("IDC").

REALTIME DATA, LLC V CME GROUP, INC., 2013-1097;

REALTIME DATA, LLC V THOMSON REUTERS CORP., 2013-1098;

REALTIME DATA, LLC V MORGAN STANLEY, 2013-1099;

REALTIME DATA, LLC V CME GROUP, INC., 2013-1100;

REALTIME DATA, LLC V THOMSON REUTERS CORP., 2013-1101;

REALTIME DATA, LLC V MORGAN STANLEY, 2013-1103.

I. STATEMENT OF JURISDICTION

The U.S. District Court for the Southern District of New York had jurisdiction over the actions giving rise to these appeals under 28 U.S.C. §§ 1331 and 1338(a). The U.S. Court of Appeals for the Federal Circuit has jurisdiction over these appeals under 28 U.S.C. § 1295(a). The notices of appeal from the final judgments entered on November 9, 2012 were timely filed under FED. R. APP. P. 4(a) and 28 U.S.C. § 2107(a) on November 21, 2012. JA1-24; JA4900-17.

II. STATEMENT OF THE ISSUES

Issue 1: Whether the district court erred in granting summary judgment of noninfringement on the basis of three misconstrued terms, collectively covering every asserted claim—"descriptor indicates," "data field/block type," and "data stream"—when, in construing those terms, the court imported extraneous limitations into the claims, disregarded an express definition in the specifications, relied on extrinsic evidence inconsistent with the intrinsic evidence, and created conflict with the constructions of other courts addressing one of the same terms in related patents.

Issue 2: Whether the district court further erred in granting summary judgment of noninfringement on the basis of those three terms when, in applying its constructions to the accused products, the court resolved factual disputes that should have been left for the jury, discounted Realtime's extensive evidence of

infringement, failed to draw reasonable inferences in Realtime's favor, and further narrowed its own constructions so as to exclude preferred embodiments.

Issue 3: Whether the district court erred in granting summary judgment of noninfringement on claims 95, 97, 108, and 112 of U.S. Patent No. 7,777,651 (the "'651 patent") when the court reached this judgment by applying an agreed construction in a manner that imported an *encoding* requirement into these *decoding* claims and, again, excluded the preferred embodiment.

Issue 4: Whether the district court erred in granting summary judgment of invalidity for indefiniteness and lack of written description on claims 1, 4, 6, 7, and 12 of the '651 patent, and claims 1, 7, 8, and 13 of U.S. Patent No. 7,714,747 ("the '747 patent"), when the purportedly invalidating terms—"content dependent data decompression" and "content independent data decompression"—have an easily-grasped meaning to persons skilled in the art, long discerned by all Defendants-Appellees ("Defendants"), and fully supported by the written descriptions.

Issue 5: Whether the district court abused its discretion in precluding Realtime from asserting infringement under the doctrine of equivalents when Realtime disclosed its assertions in compliance with the local rules and Defendants failed to show any prejudice in responding to Realtime's assertions.

III. STATEMENT OF THE CASE

In the early 2000s, foreseeing significant increases in the volume of financial market data, Realtime created and prototyped transformative data compression technology for the financial industry. JA508(2:27-35). When dramatically increased trading volumes eventually "disrupt[ed market participants'] ability to trade" and caused then-existing networks to reach a "saturation point," many industry players adopted Realtime's data compression technology. JA528; JA5210. This technology is described and claimed in the three patents-in-suit: the '651 and '747 patents, as well as U.S. Patent No. 7,417,568 (the "'568 patent").[4] JA409-524; JA5037-42.

In 2002 and 2003, Realtime successfully demonstrated its data compression technology to potential licensees in the financial industry—including certain Defendants. JA4757-85. NASDAQ, for example, concluded: "the results are impressive. [Realtime's] technology achieves 9x data compression with incredible speed." JA4884-85. Nevertheless, neither NASDAQ nor any other Defendant elected to license Realtime's technology.

Shortly thereafter in 2004, to address the "explosion" in financial market data, an industry-backed consortium known as Fix Protocol Limited developed and

[4] The three patents-in-suit are related: the '568 and '651 patents issued from the same application and share identical specifications, and both of these patents incorporate by reference the specification of the '747 patent. JA409-524.

published a financial data compression standard, known as "FAST," that incorporates Realtime's technology.[5] JA5210; JA526-31. The FAST standard is not mandatory. JA5163. Nevertheless, many in the industry have adopted FAST and recognized its substantial benefits—the very benefits provided by Realtime's technology—and have elected to use the FAST standard in their compression and decompression systems and products for, among other things, trading on stock, options, and futures exchanges. JA1749-50; JA5163.

In July 2009, Realtime brought patent-infringement suits against a number of these companies in the Eastern District of Texas. JA4918-48. Realtime's suits grouped Defendants into three categories: **1)** stock exchanges that generally encode and compress, as well as receive, decode and decompress, financial market data—BATS, CME, ISE, NASDAQ, NYSE, and OPRA (the "Exchange Defendants"); **2)** banks that generally receive, decode, and decompress financial market data from stock exchanges and then trade on that data—Credit Suisse, Goldman Sachs, HSBC, J.P. Morgan, and Morgan Stanley; and **3)** market data providers that generally receive, decode, and decompress financial market data from stock exchanges and then compile and consolidate that data for transmission to their customers—Bloomberg, FactSet, IDC, and Thomson Reuters. JA101. Every

[5] Realtime provided a visual illustration of FAST in its video tutorial submitted in advance of the *Markman* hearing. JA525.

Defendant is thus accused of infringing the patents-in-suit through the use of FAST-compliant products to compress and/or decompress financial market data.[6] JA1749-50.

In October 2011, these actions were transferred from Texas to the Southern District of New York. Thereafter, the district court summarily disposed of every asserted claim against every product accused of infringement in these actions.[7] First, on June 22, 2012, the court issued its *Markman* order, which uniformly adopted Defendants' proposed constructions for the disputed terms. JA25-65; *Realtime Data, LLC v. Morgan Stanley*, 875 F. Supp. 2d 276 (S.D.N.Y. 2012). On June 27, the court issued a summary judgment order finding claims 1, 7, 8, and 13 of the '747 patent, and claims 1, 4, 6, 7, and 12 of the '651 patent, invalid for failure to comply with the definiteness and written description requirements of 35 U.S.C. § 112. JA66-86; *Realtime Data, LLC v. Morgan Stanley*, No. 11-CV-6696-KBF, 2012 U.S. Dist. LEXIS 89561 (S.D.N.Y. June 27, 2012). Next, on August 2, the court issued an order precluding Realtime from asserting infringement under

[6] Two of the patents-in-suit, the '651 and '747 patents, issued after the original cases were filed. JA436; JA493. Realtime filed new actions under each patent against each of the three Defendant groups, and these six new actions were consolidated with the three original cases for discovery and pretrial proceedings. JA2. This accounts for the nine separate case numbers, and nine separate notices of appeal, all of which are consolidated for purposes of briefing in this Court.

[7] The asserted claims include claims 15, 20, 22, and 32 of the '568 patent; claims 1, 7, 8, 13, 14, and 19 of the '747 patent; and claims 1, 4, 6, 7, 12, 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108, and 112 of the '651 patent. JA1-24.

the doctrine of equivalents. JA87-99; *Realtime Data, LLC v. Morgan Stanley*, No. 11-CV-6696-KBF, 2012 U.S. Dist. LEXIS 109954 (S.D.N.Y. Aug. 2, 2012).

Then, on September 24, 2012—about two months before the scheduled trial, without hearing oral argument, and notwithstanding substantial evidence submitted by Realtime and its expert—the district court issued an order granting summary judgment of noninfringement as to every accused product and with respect to every asserted patent. JA100-47; JA166-213; *Realtime Data, LLC v. Morgan Stanley*, No. 11-CV-6696-KBF, 2012 U.S. Dist. LEXIS 136511 (S.D.N.Y. Sept. 24, 2012) (amended Nov. 15, 2012 to correct a typographical error). Just over one month later, on November 9, the court issued supplemental summary judgment orders ensuring that its holdings were consistently applied to all similarly situated Defendants. JA161-65. This included an order granting summary judgment of noninfringement as to NYSE and OPRA with respect to the ’568 patent. JA148-60; *Realtime Data, LLC v. Stanley*, No. 11-CV-6696-KBF, 2012 U.S. Dist. LEXIS 161436 (S.D.N.Y. Nov. 9, 2012). On that same day, the court entered final judgments under FED. R. CIV. P. 54(b) in all these actions. JA1-10; JA11-18; JA19-24.

Realtime filed its notices of appeal to this Court on November 21, JA4900-17, and these actions were consolidated for briefing in this Court on December 28, D.I.32(13-1092). To facilitate the Court’s disposition of the issues raised in these

appeals, Realtime offers a chart on page 63 (referenced in this brief as the "Issues Chart"), which cross-references each of the asserted claims with the corresponding judgments that are being challenged.

IV. STATEMENT OF FACTS

A. Realtime Invents Transformative Data Compression Technologies for the Financial Industry.

Realtime was co-founded in 1998 by James Fallon, the company's Chairman and the lead inventor of the patents-in-suit. JA409; JA436; JA493; JA5037-49. Realtime creates products and conducts research and development in various data compression technologies. JA4918. The company currently owns more than twenty issued U.S. patents and has numerous pending patent applications. JA5039.

Fallon and his co-inventors recognized early on that their data compression technologies were transformative for the financial industry—an industry that was experiencing tremendous growth in trading volume and associated transmission of market data. JA4757-67.

The inventors recognized that in prior art compression systems, "data is often segregated into packets for transmission." JA509(3:51-52). They also recognized that "[b]y aggregating the data [packets], and then applying compression, somewhat higher compression ratios are often achieved. This then translates to lower data communications costs or more customers served for a given amount of available communications bandwidth." JA509(3:57-62). But the

inventors went against that teaching in the art and conceived of a system that, among other things, applied what they called "Packet Independence," which they defined as a compression scheme that "has no packet-to-packet data dependency." JA511(8:6-7); JA422(7:62-66). In other words, their concept did not use or require, and in fact avoided, any "aggregation" of data packets to achieve increased compression as taught and practiced in the prior art. That "Packet Independent" approach to compression also reduced the "latency," or time delay, that results from having to wait until data in multiple packets can be aggregated before compressing or decompressing the data. JA509(3:62-65).

"Packet Independent" data compression was not the inventors' only contribution to the art. Realtime conceived of a complimentary technology called "content dependent" data compression, which allowed for significant compression of data even when applied to only the relatively small amount of data in a single data packet, which in financial data packets is typically about 1,000 bytes. JA487-88(15:50-26:3); JA514-15(14:35-15:3); JA546; JA1877. Content dependent compression, as described in the patents-in-suit, takes advantage of "a-priori knowledge of the structure and content" of a data stream. JA512(9:17-21,11:46-50). Realtime recognized that if it had prior knowledge about a data stream, such as the likely content or structure of the fields in each packet, or any other characteristic or parameter of the data fields, it could design a list or computer file

(also called a “state machine”) based on “the structure and content” of the data stream. JA512(9:16-21); JA486(16:12-52); JA520(25:2). That list associates a highly optimized encoder with each data field, which encoder is selected to encode the data field if the expected characteristic or parameter is met. JA518(22:39-23:35); JA514(14:51-55). Realtime’s approach took advantage of “structured data streams having repetitive data content (e.g., stock symbols and quotes, etc.).” JA513(11:45-50).

Before an associated encoder is applied to data in a data block (or data field), Realtime’s claimed inventions analyze the content of the data block to determine the “characteristics of the data block.” JA480(4:27-32). If the data block has the expected characteristic or parameter associated with the encoder, that encoder is applied. This is what the patents-in-suit refer to as “content dependent” data compression. If, however, the data block does not have the expected characteristic or parameter associated with the content dependent encoder, another encoder is applied to ensure at least some compression. JA461-62. The patents-in-suit refer to this latter approach as “content independent” data compression. JA480(3:43-56).

Financial data is typically transmitted between financial institutions—where one system (*e.g.*, an exchange) generates, compresses, and transmits the data, while another system (*e.g.*, a bank) receives and decompresses it. JA413-14. Because decompression must necessarily reverse the effects of compression, an effective

compression system must provide some means for the decompressing system to know which encoders were selected to compress the different fields in the transmitted data packet. JA1209-10. Realtime's technology solves this problem through the use of data compression type "descriptors," defined broadly in the patents as "any recognizable data token … that indicates which data encoding technique has been applied to the data." JA515(16:23-26); JA482(8:53-56). An example of such a descriptor comprises one or more data tokens that indicate the list, computer file, or state machine that indentifies the associated encoders for each data field (referred to in FAST as a "template"), and whether content dependent compression (the associated encoder) or content independent compression (another encoder for at least some compression) was used to encode the data field (referred to in FAST as a "Presence Map" or "PMAP"). JA518(22:44-46); JA512(9:16-21); JA520(25:2); JA486(15:31-37); JA485(14:54-59); JA486(15:50-60); JA532-549.

These concepts are reflected in, for example, asserted encoding claims 22 and 25 of the '651 patent, JA520, and claim 32 of the '568 patent, reproduced below:

> 1. A method for compressing data, wherein one or more types of encoding are applied to a data stream depending on identifiable data fields in the data stream, the method comprising:
>
> recognizing a data field type of a data field in the data stream, wherein the data field is included in a packet;

> selecting an encoder associated with the recognized data field type;
>
> encoding the data in the data field with the selected encoder; and
>
> providing a descriptor with the encoded data which identifies the selected encoder.

…

32. The method of claim 1, wherein the encoding does not require packet-to-packet data dependency.

JA430-31. In addition, claim 20 of the ’568 patent recites a “data feed dependent data compression” method that “creat[es] a description file” that corresponds to the computer file or “state machine” model described in the specification. JA430; JA423(9:11-29,10:17-22). As shown below, the accused instrumentalities practice such a “state machine” model through the use of “templates.” Other asserted claims teach decoding systems and methods that reverse the encoding performed with this technology, using one or more descriptors. JA491-92(claims 1,8); JA519-24(claim 108).

B. Financial Industry Players Find Realtime’s Technology Provides “Impressive” Results with “Incredible Speed,” but Elect Not to Take Licenses.

Realtime did more than apply for patents on its technology—it built a prototype system and demonstrated that prototype to some of the Defendants in this action, specifically NASDAQ and Thomson Reuters. Thomson Reuters and NASDAQ both paid Realtime for these demonstrations, concluding—as NASDAQ’s CTO put it—that “the results are impressive. [Realtime’s] technology

achieves 9x data compression with incredible speed." JA4884-85; JA4808; JA4894-95; JA4791-92. Notwithstanding these impressive results, neither NASDAQ nor Thomson Reuters elected to license Realtime's technology.

Moreover, in 2002, Fallon wrote a white paper explaining how Realtime's compression concepts could benefit the financial data market. JA4757-67 ("Accelerated Data Delivery in Financial Information & Transaction Networks"). This paper was obtained by some Defendants, including [redacted]. JA4768-80.

C. The FAST Standard Incorporates Realtime's Technology.

Not long after Realtime demonstrated its technology to potential licensees, an industry-supported organization known as FIX (*F*inancial *I*nformation e*X*change) Protocol Limited ("FPL") created a working group to address the same problem that Realtime and Fallon had addressed: how to handle dramatic increases in the volume of financial market data using existing bandwidth while minimizing latency. JA526-31; JA5207-5227. The standard that the FPL group created to solve this problem—known as FAST, or *F*IX *A*dapted for *St*reaming—incorporates Realtime's technology. JA532-49; JA5043-47.

Like Realtime's technology, for example, FAST takes a packet independent approach. The FAST data stream segregates data into packets, which contain a sequence of one or more blocks or messages, which are in turn made up of data fields. JA1756; JA1811-12. Just as taught in Realtime's patents, the packets are

"encoded completely independently of each other and **no <u>compression state [is] kept in between packet transmissions</u>**." JA1879(emphasis in original); JA546 ("A FAST implementation should be configured to process no more than a single [packet] due to the unreliable nature of [packet-switched networks].").

Like Realtime's technology, FAST also relies upon *a priori* knowledge of the structure and content of the data stream. As the FAST user's guide explains, "[t]he FAST protocol is predicated on familiarity with the content of a [financial market] data feed." JA536. This means that, as taught in Realtime's patents, the FAST standard can employ "templates which describe the content and characteristics of the data to be encoded or decoded … and instruct FAST how to encode or decode each field within a message." JA536; JA539; JA518(22:44-46); *see also* JA512(10:28-33) ("the present invention is applicable with any data stream whose statistical regularity may be captured … in a state machine model"); JA539 ("FAST functions as a state machine"). In particular, FAST associates three different data field types with three different optimal encoders: a "redundant" data field type is associated with a copy encoder; a "sequentially different" data field type is associated with an increment encoder; and a "most common value" data field type is associated with a default encoder. JA1752; JA1760. Thus again, as taught in Realtime's patents, FAST compresses different fields in a data packet

with different encoders, selecting the optimal encoder for each field. JA1751-53; JA536.

Like Realtime's technology, FAST also employs "content dependent" and "content independent" compression. FAST, for example, will analyze the content of each data field and determine (or check) whether that content has the relevant characteristic or parameter (*i.e.*, data field type)—redundant, sequentially different, or most common value—for one of the associated encoders. JA1751-53. If FAST identifies a relevant data field type, the data in that field will be compressed with the associated optimal encoder. A redundant type thus will be compressed with copy encoding. JA1751-52. This is content dependent data compression. If, on the other hand, FAST does not identify a relevant data field type, the data in the field will be compressed with another encoding technique selected for this purpose to provide at least some compression—for FAST systems, stop-bit encoding. JA1753. This is content independent data compression.

Like Realtime's technology, FAST also makes use of descriptors to indicate the different encoding techniques selected to compress different data fields. In particular, a "Presence Map" ("PMAP") and "Template ID" are included as a descriptor with each message in a FAST data packet.[8] JA1755-56. The Template

[8] The function of the Template ID is sometimes performed by a "message type" or "message category" in the accused systems. JA1755-56.

ID identifies and provides a link to the correct template—an *a priori* list of the optimal encoders (copy, increment, or default) associated with each data field. JA539-47; JA1860-62; JA119; JA1845. The PMAP indicates, for each field in the message, whether the associated optimal encoder (content dependent compression) or the stop-bit encoder (content independent compression) was used to compress the data in that field. JA1753-56; JA538; JA1812. Thus, as taught in Realtime's patents, FAST uses "recognizable data token[s] … indicat[ing] which data encoding technique has been applied to the data." JA515(16:23-26); JA1755-56.

Finally, like Realtime's technology, FAST systems employ decoding techniques that reverse the effects of the corresponding encoding techniques. For example, FAST decoding systems use a descriptor (the Template ID and PMAP) to determine the applied encoder and whether a given data field was encoded with either content dependent or content independent data compression, and then apply the corresponding decoding technique. JA1762-69; JA545.

D. Realtime Brings Suit; the District Court Grants Summary Judgment on Every Asserted Claim.

Many in the financial industry have recognized FAST's substantial benefits—the very benefits provided by Realtime's technology—and have elected to use the FAST standard in their compression and decompression systems. JA1749-50; JA5163. In 2009, Realtime brought patent-infringement suits against a number of these companies in the Eastern District of Texas. JA4918; JA4929;

JA4939. These suits were transferred to the Southern District of New York in October 2011, JA4949-52, and assigned to Judge Forrest in February 2012, JA4953-54.

The district court held a *Markman* hearing a few months later, and issued its order construing the disputed claim terms in June 2012. JA63-65. The court adopted each of Defendants' proposals. JA25-65. This required, however, importing limitations into the claims, ignoring a definition in the specifications, relying on extrinsic evidence, and creating conflict with the constructions of other courts addressing one of the same terms in related patents.

Less than a week after its *Markman* opinion, the district court issued an order finding nine of the asserted claims invalid for failure to comply with the definiteness and written description requirements. JA85. These claims contained the terms "content dependent" and "content independent … decompression"—which the court, relying on extrinsic evidence from Defendants' expert, found meaningless and impossible to construe. JA84. The relevant intrinsic evidence, however—in particular, the reexamination proceedings on the patents-in-suit—established that these decompression terms are naturally understood by persons skilled in the art (and by Defendants) to mean reversing "content dependent" and "content independent … compression." JA1218-21; JA1229-33.

Shortly thereafter, in August 2012, the court issued an order precluding Realtime from asserting infringement under the doctrine of equivalents (or "DOE"). JA99. Realtime's infringement contentions had raised DOE allegations, and Defendants never moved for more detailed DOE contentions. JA4458-61. The court nonetheless found that Realtime had not preserved its DOE claims. JA94.

Then, on September 24, 2012—about two months before the scheduled trial, and without hearing oral argument—the district court granted broad summary judgments of noninfringement, finding that all of the accused products failed to meet one or more of the elements of each asserted claim of the patents-in-suit. JA100-47. The court found, for example, that the accused products failed to meet the "descriptor indicates," "data field type," and "data stream" limitations as those terms had been construed. JA117-20; JA124-28; JA133-46. The court thus determined that there were no material factual disputes as to these limitations. JA118-145. Realtime's evidence, however, demonstrated that: the "PMAP" and "Template ID" used in the FAST systems satisfy the "descriptor" limitation as construed by the court; the "redundant," "sequentially different," and "most common value" data types used in the FAST systems satisfy the "data field type" limitation as construed by the court; and the "financial data stream" used in the FAST systems satisfies the "data stream" limitation as construed by the court. JA1746-91.

The district court also granted summary judgment of noninfringement on certain decoding claims after concluding that the accused *decoding* products failed to meet an *encoding* requirement that, the court concluded, had been imported into these decoding claims through an agreed construction. JA130-31; JA1406. This agreed construction, however, was merely intended to clarify a "wherein" clause that describes the characteristics of the received, encoded data packet (*i.e.*, the encoding technique used) so that it can be properly decoded. JA523-24(claim 18). And applying the construction as the court did is nonsensical in the context of the claim language and excludes a preferred embodiment. The specifications of the patents-in-suit expressly envision that one party (encoding system) may compress and transmit financial data, while another party (decoding system) receives and decompresses that data. JA502-03; JA510(5:47-54); JA511-12(8:46-9:21).

On November 9, 2012, following the entry of its wide-ranging summary judgment orders in September, the district court entered supplemental orders ensuring that its holdings were consistently applied to those Defendants who had joined in the various and respective summary judgment motions. JA148-65. The court then entered final judgments against Realtime and in favor of Defendants, and these appeals followed. JA1-24; JA4900-17.

V. SUMMARY OF ARGUMENT

A. The district court erred in granting summary judgment of noninfringement on the basis of three misconstrued claim terms that, collectively, cover every asserted claim. In construing these three terms—"descriptor indicates," "data field/block type," and "data stream"—the court imported extraneous limitations into the claims, ignored an express definition in the specifications, relied on extrinsic evidence inconsistent with the controlling intrinsic evidence, and created conflict with the constructions of two other courts addressing related patents. Because these three terms were erroneously construed, the summary judgment orders based upon them should be reversed.

B. The district court further erred in granting summary judgment of noninfringement on the basis of these terms because, even as they were construed, Realtime proffered evidence sufficient to raise genuine disputes of material fact with respect to infringement of the asserted claims. This evidence included FAST documentation, a declaration from Realtime's expert, and more than 2,000 pages of detailed infringement charts pointing to specific lines of code to show infringement by each of the accused products. In granting summary judgment, the district court improperly resolved factual disputes that should have been left for the jury, failed to draw reasonable inferences in Realtime's favor, and applied one of

its constructions—that of "data stream"—so as to exclude the preferred embodiments.

C. The district court also erred in granting summary judgment of noninfringement as to decoding claims 95, 97, 108, and 112 of the '651 patent on the basis of an agreed construction. That construction merely explains a "wherein" clause of Realtime's "decoding" claims, which describes the characteristics of the received, encoded data packet. The district court nonetheless applied the agreed construction to import a compression (or *encoding*) limitation into these decompression (or *decoding*) claims. It applied the construction, that is, to require that the same party both compress and decompress the financial data stream. This excludes the preferred embodiment, which teaches separate encoding and decoding systems, and makes little sense in the context of a "decoding" claim. These orders should also be reversed.

D. The district court also erred in granting summary judgment of invalidity as to decoding claims 1, 4, 6, 7, and 12 of the '651 patent and decoding claims 1, 7, 8, and 13 of the '747 patent. The court found that it made good sense to describe certain encoding techniques as "content dependent" and "content independent," but that it made no sense to describe the corresponding decoding techniques that reverse the encoding techniques with corresponding language. Relying on extrinsic evidence, the court found, that is, that the terms "content

dependent data decompression" and "content independent data decompression" had no meaning, were impossible to construe, and thus failed the definiteness and written description requirements of 35 U.S.C. § 112. This conclusion, however, conflicts with the evidence, including the intrinsic evidence—which shows that both sides understand these decompression terms to signify the counterparts of their corresponding compression terms. Indeed, Defendants revealed no confusion over the bounds of these claim terms during reexamination proceedings or when formulating invalidity contentions. Because these decompression terms can be given a reasonable meaning, and because that meaning is supported by the written description and intrinsic evidence, the district court's summary judgment order should be reversed.

E. Finally, the district court abused its discretion in precluding Realtime from asserting infringement under the doctrine of equivalents. Realtime complied with the applicable local rules and missed no court-ordered deadlines: it timely disclosed its DOE allegations in its infringement contentions, and it served an expert report opining on those allegations—in the particular context of Defendants' proposed claim constructions—a week before the district court issued its *Markman* ruling. Defendants never moved for more detailed DOE contentions, and demonstrated no prejudice in responding to the DOE-related opinions of

Realtime's expert. The court's order excluding those opinions, and precluding Realtime from asserting infringement under the DOE, should thus be reversed.

VI. STANDARDS OF REVIEW

The district court's claim constructions and its summary judgment orders on noninfringement and invalidity are all subject to *de novo* review in this Court. *Cybor Corp v. FAS Techs., Inc.*, 138 F.3d 1448, 1454-55 (Fed. Cir. 1998) (en banc) (claim construction); *Praxair, Inc. v. ATMI, Inc.*, 543 F.3d 1306, 1319 (Fed. Cir. 2008) (indefiniteness); *ICU Med., Inc. v. Alaris Med. Sys.*, 558 F.3d 1368, 1376 (Fed. Cir. 2009) (written description); *Schindler Elevator Corp. v. Otis Elevator Co.*, 593 F.3d 1275, 1281 (Fed. Cir. 2010) (noninfringement).

The summary judgment orders can stand only if the record shows "that there is no genuine dispute as to any material fact and [that Defendants are] entitled to judgment as a matter of law." FED. R. CIV. P. 56(a). In reviewing those orders and the underlying evidence, this Court "resolv[es] reasonable factual inferences in favor of the patentee." *Absolute Software, Inc. v. Stealth Signal, Inc.*, 659 F.3d 1121, 1130 (Fed. Cir. 2011). In other words, Realtime's summary judgment evidence "is to be believed, and all justifiable inferences are to be drawn in [its] favor." *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 255 (1986). The court's invalidity-related summary judgment orders on indefiniteness and the written description requirement can further stand only if, when viewed in the light most

favorable to Realtime, those orders are supported by clear and convincing record evidence. *ICU Med.*, 558 F.3d at 1376.

The district court's decision to preclude Realtime from asserting infringement under the doctrine of equivalents is reviewed for an abuse of discretion. *Mintz v. Dietz & Watson, Inc.*, 679 F.3d 1372, 1375 (Fed. Cir. 2012). A court "abuses its discretion when its decision is based on clearly erroneous findings of fact, is based on erroneous interpretations of the law, or is clearly unreasonable, arbitrary or fanciful." *Cybor*, 138 F.3d at 1460.

VII. ARGUMENT

A. The District Court Erred in Granting Summary Judgment of Noninfringement on the Basis of Three Misconstrued Terms That, Collectively, Cover Every Asserted Claim.

1. The court misconstrued "descriptor indicates."

The term "descriptor indicates" appears in every asserted claim of the '651 and '747 patents and in asserted claims 15 and 32 of the '568 patent.[9] *See* Issues Chart. Claim 43 of the '651 patent, for example, claims a method "comprising: … providing a descriptor for the encoded data packet, wherein the descriptor indicates the selected one or more lossless encoders for the encoded data block." JA521.

[9] The term is sometimes plural: "descriptors indicate." JA64; JA491. Two claims use a related term: "descriptor with the encoded data which identifies." JA430(23:44-45,24:36).

Realtime proposed that the term be construed simply as a "data token" that, tracking the relevant claim language, "indicates" the selected encoder. JA61. This construction reflects the express definition of the term in the specifications: "A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data." JA515(16:23-26); JA482(8:53-56).

The district court rejected Realtime's proposal, and instead adopted the one offered by Defendants: "recognizable data that is appended to the encoded data for specifying" the selected encoders. JA61. In so doing, the court committed two fundamental errors.

First, the court adopted a construction at odds with the provided definition. When a "patentee explicitly defines a claim term in the patent specification, the patentee's definition controls." *Martek Biosciences Corp. v. Nutrinova, Inc.*, 579 F.3d 1363, 1380 (Fed. Cir. 2009). Realtime pointed the district court to the express definition for this term in the specifications, but the court failed to apply it. JA61-63; JA1450. That was error. *Martek*, 579 F.3d at 1380.

Second, the court adopted a construction that imported two extraneous limitations into the claim term: "appended to" and "for specifying." That was also error. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005).

With respect to the "appended to" limitation, the district court noted that the specifications suggest—in describing the preferred embodiments—that descriptors can be "append[ed]" to data fields and data blocks. JA61-62. A proper construction, however, "must not import limitations from the specification into the claims." *Energy Transp. Group, Inc. v. William Demant Holding A/S*, 697 F.3d 1342, 1349 (Fed. Cir. 2012). And while two of the asserted claims require "providing a descriptor with the encoded data," JA430(23:44-45,24:36), none require that the descriptor be "appended to" the data. JA894-96. The court erred in concluding otherwise. JA64.

With respect to the "for specifying" limitation, the district court provided no explanation or analysis. JA60-63. And the intrinsic evidence provides no basis for importing this limitation into the claims. The claims themselves—tracking the definition provided in the specification—state that the descriptor "indicates" (or, in two cases, "identifies") the selected encoders. JA515(16:23-26); JA430(23:44-45,24:36); JA520(25:16). Swapping "specifies" for "indicates" and "identifies" provides no additional clarity to these commonly understood words. JA896. It does, however, improperly narrow the claims. *See Phillips*, 415 F.3d at 1314, 1323.

2. The court misconstrued "data field/block type."

The terms "data field type" or "data block type" appear in, or are incorporated into, every asserted claim of every asserted patent. *See* Issues Chart.

Claim 20 of the ’568 patent, for example, requires “recognizing data field types in the data stream and applying encoders associated with the recognized data field types to encode the data stream.” JA430.

Realtime proposed that these terms be construed as “an attribute or characteristic of the data block or data field.” JA47. This is consistent with the specifications, pursuant to which a “data field/block type” is any characteristic, attribute, or parameter of the data field or block that is used to select an appropriate encoder for that field or block. JA489-90(22:64-23:1) (“analyze the incoming data stream for recognition of data types, data str[u]ctures, data block formats, file substructures, file types, *or any other parameters* that may be indicative of the appropriate data compression algorithm”) (emphasis added); JA480(4:30-32) (“estimating a desirability of using one or more encoder types based on characteristics of the data block”); JA486(16:20-21) (“associations between recognized data parameters and appropriate algorithms”); JA518(22:44-45) (“possible data fields and parameters”). The patents envision no limitation on the kind of characteristics, attributes, or parameters that might be used to select the appropriate encoders. JA489-90; JA5078-80.

The district court nevertheless rejected Realtime’s proposal, and instead adopted the one offered by Defendants: “categorization of the data in the field (or block) as one of ASCII, image data, multimedia data, signed and unsigned

integers, pointers, or other data type." JA48. Citing to extrinsic evidence improperly credited to Defendants' expert Dr. Storer, the court further narrowed this construction by holding that "checking a value is not determining [or recognizing] a data block type or a data field type." JA50. In so ruling, the district court again committed two fundamental errors.

First, the court again adopted a construction that improperly narrowed the broad claim language based on extrinsic expert testimony, and imported extraneous limitations from the specifications into the claims. *Energy Transp. Group*, 697 F.3d at 1349.

In particular, the construction's catalog of "ASCII, …," etc. is drawn from a list of exemplary data types found in the specification. JA514(14:20-22). There is plainly no reason to import a requirement for the use of one of those specific data types into the claims. *Phillips*, 415 F.3d at 1323. Recognizing this, the district court suggested that adding "or other data type" to the construction permitted the "term to include data types that might not be listed but would be fairly encompassed by the invention." JA48. If the exemplary list is ignored, therefore, the court effectively construed "data field/block type" to mean a "data type … fairly encompassed by the invention." Aside from the obvious uncertainty—and circularity—inherent in defining a term to cover those data types that are "fairly encompassed by the invention," the construction equates "data field/block type"

with "data type." But the inventors used the term "data field/block type" to signify a scope broader than the term "data type." JA888-889.

And when the court's exemplary listing of "ASCII, …," etc. is considered, the error is compounded—for then the construction suggests that the only data types "fairly encompassed by the invention" are data format characteristics. JA48; JA514. The patents are clear, however, that the invention is not limited to the determination of data formats. JA888-89; JA489-90(22:64-23:1). Not only do the patents list a data format as just *one* example of a characteristic that can be associated with an encoder, JA489-90(22:64-23:1), but also the specification for the '747 patent, in fact, explains that the required identification of particular data formats such as "ASCII, binary, or unicode" was a disadvantage inherent in the prior art, JA480(3:21-36) (describing limitations associated with the Chu method). The court's construction, therefore—focusing as it does on data formats such as ASCII—effectively narrows the claims to an embodiment that the specification describes as disadvantageous. JA48; JA480(3:21-36); JA888-889.

Second, in reaching its construction of this term—and its further narrowing of that construction—the district court improperly relied on extrinsic noninfringement-driven testimony and argument from Defendants' expert and counsel. The law is clear that claims may not be construed with "the objective of capturing or excluding the accused device." *Vita-Mix Corp. v. Basic Holding, Inc.*,

581 F.3d 1317, 1324 (Fed. Cir. 2009). And while “trial courts generally can hear expert testimony for background and education on the technology,” *Key Pharms. v. Hercon Labs. Corp.*, 161 F.3d 709, 716-17 (Fed. Cir. 1998), they must refrain from “crediting certain evidence over other evidence” during claim construction, *Cybor*, 138 F.3d at 1455 n.5, and from relying on extrinsic evidence “clearly at odds” with the intrinsic evidence, *Phillips*, 415 F.3d at 1318. The district court’s construction ran afoul of these rules.

Both sides offered expert testimony in support of their proposals: Realtime’s expert, Dr. Shamos, testified that Defendants’ proposal was too narrow; Defendants’ expert, Dr. Storer, testified that it was not. JA888-889; JA49. The court improperly credited the opinion of Dr. Storer over the opinion of Dr. Shamos in finding that “[D]efendants’ proposed construction is the proper one.” JA49. The court also improperly credited Dr. Storer’s testimony as providing the basis for its further conclusion that “checking a value is not determining a data block type or data field type.” JA50. That in itself would have been improper, but the error was again compounded. The relevant testimony did not in fact come from Dr. Storer on claim construction—it came from Defendants’ counsel at the *Markman* hearing:

> MR. HAWES: Your Honor, I’m going to be up front with you. This is about noninfringement. … [W]e, the defendants, are here to give you positions that help us. … We need to know whether just checking a value is going to have the impact of determining the data type.
> …

> THE COURT: I'm trying to figure out what the infringement contention means.
> …
> MR. HAWES: … The way they are trying to argue infringement for determining the data type is looking for a specific value and saying that that is determining the data type.

JA793-98.

Now to be sure, Defendants mischaracterized Realtime's infringement arguments. But that does not change the fact that they invited the court to construe this term with the objective of precluding those arguments, and the court accepted that invitation. JA50. As this Court has held, such an approach "make[s] infringement a matter of judicial whim." *Sri Int'l v. Matsushita Elec. Corp.*, 775 F.2d 1107, 1118 (Fed. Cir. 1985). Nothing in the intrinsic record disclaims "checking a value" as a way to determine a data field type, and the district court erred in concluding otherwise. JA793; JA50.

3. The court misconstrued "data stream."

The term "data stream" (or "stream of data") also appears in every asserted claim. *See* Issues Chart. Claim 14 of the '747 patent, for example, requires "providing a descriptor for the compressed data packet in the data stream" being transmitted. JA492.

Realtime proposed that the term be construed as "one or more data blocks transmitted in sequence." JA37. This is consistent with the specifications, which describe a "data stream comprising one or more data blocks [that] is input into the

data compression system." JA482(8:3-7); JA512(10:28-33); JA513(11:45-50). Indeed, there was no dispute that this proposal is consistent with the claims and specifications of the patents-in-suit. JA37-38.

The district court nevertheless rejected Realtime's proposal, and adopted the one urged by Defendants. JA37. That construction began as Realtime proposed, but continued with additional limitations not inherent in the ordinary meaning of the term: "one or more data blocks transmitted in sequence *from an external source whose characteristics are not controlled by the data encoder or decoder*." JA37(emphasis added). The court offered two reasons in support of its decision to import these narrowing limitations into the claim language. JA38-43. Neither has merit.

First, the court pointed to a declaration filed on behalf of Realtime during reexamination of a patent not at issue here, but sharing the specification of the '747 patent: U.S. Patent No. 7,161,506. JA39. The excerpt highlighted by the court indicated that

> [a] person of ordinary skill in the art would consider the phrase 'receiving a data stream' to imply a stream of data transmitted from a source (whose characteristics are therefore not controlled by the data compression system) and received at the input of a system or device.

JA40. The court found no legal disclaimer based on this declaration, but it nonetheless determined that the claims, the specifications, and Realtime's arguments in this litigation were all trumped by these "factual positions made by

those skilled in the art." JA41. The court found, that is—without citation to any authority—that the "assertions in declarations of related patents are alone sufficient to support adoption of [D]efendants' construction." JA41-42.

This finding was in error for numerous reasons. **1)** The law holds that limitations cannot be imported into claim terms solely on the basis of, in the court's words, "factual positions made by those skilled in the art." JA41; *Key Pharms.*, 161 F.3d at 716 ("[claim meaning] cannot be altered or superseded by witness testimony"). **2)** Significantly, the court's highlighted excerpt addressed a narrower term—"*receiving* a data stream"—and the same declaration elsewhere proposed a construction for the broader term "data stream" as "a continuous stream of data elements received from or transmitted." JA39-40; JA970(¶10). **3)** The court's construction was never adopted by the PTO; the examiner in fact rejected the arguments offered in the highlighted declaration excerpt. JA959-60; JA1433-36. And when the PTO "rejects an applicant's proposed construction, the idea of public notice cuts in the opposition direction"—that is, against adopting the rejected construction. *Abbott Labs. v. Church & Dwight Co.*, No. 07-C-3428, 2008 U.S. Dist. LEXIS 103635, at *24 (N.D. Ill. Dec. 22, 2008). **4)** Finally, in response to the intervening-rights decision in *Marine Polymer Techs., Inc. v. HemCon, Inc.*, 659 F.3d 1084 (Fed. Cir. 2011)—later vacated, 475 F. App'x 315 (Fed. Cir. 2012)—Realtime retracted the reexamination arguments on which the district court

relied. JA952-61; JA1433-34. The court recognized as much, but held that because there was no procedural mechanism to expunge the filed declaration, it effectively imported limitations into the claims of related patents. JA41-42. There is, however, no authority for such a rule, and the reexamination proceedings highlighted by the district court do not support its estoppel-like construction.

Second, the district court pointed to two other litigations in which claims from related Realtime patents have been construed—the *Packeteer* and *MetroPCS* litigations—and noted that, "[a]s a matter of law, claim terms appearing in related patents should generally be construed consistently across them." JA43; *Omega Eng'g Inc. v. Raytek Corp.*, 334 F.3d 1314, 1334 (Fed. Cir. 2003). However, as explained below, this rule provides a compelling reason to adopt the construction proposed by Realtime, not a reason to reject it.

In the *Packeteer* litigation, the court construed the same "data stream" term as "one or more data blocks transmitted in sequence." JA4992-93. Realtime's proposal tracks this construction. JA42. In the *MetroPCS* litigation, the court—citing *Packeteer*—construed "receiving a data stream" as "receiving from an external source one or more data blocks transmitted in sequence." *Realtime Data, LLC v. MetroPCS Texas, LLC*, No. 6:10-CV-493, 2012 U.S. Dist. LEXIS 141987,

at *31 (E.D. Tex. Oct. 1, 2012).[10] It narrowed the additional claim term "receiving," in other words, to require "receiving from an external source." The *MetroPCS* court did so because, it explained, all parties agreed that the patents at issue in that case were directed to "systems that receive data from an external source." *Id.* at *25. But neither of these courts construed the term—even with an additional "receiving" limitation—to further require "an external source *whose characteristics are not controlled by the data encoder or decoder*." *Id.* at *31; JA4992-93; JA37(emphasis added).

The only construction of the broader term "data stream" that is consistent with the constructions in the *Packeteer* and *MetroPCS* litigations, therefore, is the one offered by Realtime: "one or more data blocks transmitted in sequence."

4. The court erred in granting summary judgment of noninfringement on the basis of these misconstrued terms.

A summary judgment of noninfringement "should ordinarily be vacated or reversed" when it is "based on a claim construction that this [C]ourt determines to be erroneous." *Burke, Inc. v. Bruno Indep. Living Aids, Inc.*, 183 F.3d 1334, 1338 (Fed. Cir. 1999); *Int'l Visual Corp. v. Crown Metal Mfg. Co.*, 991 F.2d 768, 772 (Fed. Cir. 1993). As noted in the Issues Chart on page 63, the district court granted summary judgment of noninfringement based on its constructions of "descriptor,"

[10] The *Markman* proceedings in the *MetroPCS* action overlapped the *Markman* proceedings in this action, and the court in *MetroPCS* issued its *Markman* order about three months after the *Markman* order was issued in this action. JA42-43.

"data block/field type," and "data stream." Because these constructions were erroneous, the summary judgment orders flowing from them should be reversed.

B. The District Court Further Erred in Granting Summary Judgment of Noninfringement on the Basis of These Terms Because, Even as Construed, Disputes of Material Fact Remain.

Realtime requests that this Court correctly construe the terms "descriptor indicates," "data field/block type," and "data stream," and remand these cases for trial. But even if the Court finds the district court's claim constructions proper, it should nevertheless reverse the summary judgment orders and remand, because the district court committed several errors in applying its constructions to the summary judgment evidence. These include resolving factual issues that should have been left for the jury, failing to draw reasonable inferences in Realtime's favor, and failing to acknowledge evidence presented by Realtime that demonstrated genuine disputes of material facts—including FAST documentation, a declaration from Realtime's expert, and thousands of pages of infringement charts detailing how each and every element of the asserted claims is met by each of the accused products under the district court's constructions. JA526-49; JA1487-94; JA1746-91; JA2734-861; JA3767-4082.[11]

[11] The charts for ISE, NYSE, and OPRA are representative of Dr. Shamos' charts for all Defendants, which due to their length are not included in the joint appendix but are part of the record below.

1. There is a genuine dispute regarding whether the accused products meet the "descriptor indicates" limitation.

The district court granted summary judgment of noninfringement on twenty-one asserted claims for an alleged failure to meet the "descriptor indicates" limitation under the court's construction of that term as "recognizable data that is appended to the encoded data for specifying." *See* Issues Chart. The court's analysis turned on its imported "appended to" and "for specifying" limitations. JA117-20. Even accounting for these extraneous limitations, however, Realtime offered substantial summary judgment evidence—including an expert declaration, infringement charts, and supporting documents—creating, at the very least, a genuine factual dispute as to whether the accused products make use of "recognizable data that is appended to the encoded data for specifying" the selected encoders. JA1755-56; JA547; JA2741; JA3773-74. The court erred in concluding otherwise.

Realtime's summary judgment evidence demonstrated that a FAST-encoded data stream is comprised of a sequence of one or more messages or blocks in a data packet, with a "Presence Map" ("PMAP") and "Template ID" appended to each encoded message. JA547; JA1756; JA1811-12. The PMAP and Template ID are there for one reason and one reason only: "for specifying" the encoders that were used to encode the data fields. JA1756. As Dr. Shamos explained, together the

PMAP and Template ID thus function as the "descriptor" in the accused products. JA1755-56.

In particular, the PMAP specifies, for each field in the message, whether the *content dependent* encoder (the associated optimal encoder) was used to encode the data in that field. JA538; JA1755-56; JA1812. If the PMAP specifies that it was not, then the *content independent* encoder (the stop-bit encoder) was used. JA1753-56. The Template ID next specifies and provides a link to the correct template—which itself is simply a static, *a priori* list of the optimal encoders associated with each data field. JA539-47. Together, the appended PMAP and Template ID thus specify that, for example, the data in a given field was encoded with the associated encoder listed for that field in the linked template. JA1756; JA547.

The system checks the identified template to find that the encoder specified by the PMAP and Template ID happens to be, for example, the copy or incremental encoder. JA541-47. But again, the template is simply a static, *a priori* list of associations; it does not specify whether the associated encoder was actually selected to encode a given data field. As Realtime's evidence showed, the PMAP and Template ID[12] together perform that function. JA1755-56.

---

[12] As noted above, Realtime's summary judgment evidence demonstrated that the function of the Template ID is sometimes performed by a "message type" or "message category" in the accused systems. JA1756; JA1764.

The controlling summary judgment rules required the district court to credit Realtime's evidence, and to draw every reasonable inference in its favor. *Absolute Software*, 659 F.3d at 1130. The court did not do so. Instead, it credited Defendants' evidence, pointing to testimony from their fact witness to the effect that "the template ID and the PMAP do not themselves provide any indication as to which type of … encoding might have been applied to a given piece of data," and inferring from this disputed opinion that it is "the template, which in fact identifies the encoder that has been used." JA119. The court then granted summary judgment on the ground that the static template was doing the specifying, but it was not appended with the PMAP and Template ID to the encoded data. JA119-20.

The court's resolution of factual disputes against Realtime requires reversal of its summary judgment order. But it is worth further noting that the manner in which the court resolved these genuine disputes was inconsistent with the patents themselves. The patents repeatedly explain the virtue of making use of *a priori* information to create lists, computer files, or other state machines (or a "template" in FAST), to maximize compression and reduce latency in the transmission of encoded data. JA512(9:17-21,10:28-33); JA513(11:46-50); JA518(22:28-32). That is how the FAST system works with respect to the "descriptor" limitation, and to grant summary judgment on the ground that the system uses an *a priori* template that need only be sent to the decoder once—to avoid the repeated transmission of

redundant data with every message—is inconsistent with the patents and the inventions they disclose. JA118-20.

Even under the district court's construction of "descriptor," Realtime's summary judgment evidence raised, at the very least, a genuine factual dispute as to whether the PMAP and Template ID sent with every FAST-encoded message constitute "recognizable data that is appended to the encoded data for specifying." JA1755-56; JA547. The court's holding to the contrary should be reversed.

2. There is a genuine dispute regarding whether the accused products meet the "data field/block type" limitation.

The district court granted summary judgment of noninfringement on twenty-three asserted claims for an alleged failure to meet the "data field/block type" limitation under the court's construction of that term as "categorization of the data in the field (or block) as one of ASCII, image data, multimedia, signed and unsigned integers, pointers, or other data type." *See* Issues Chart. The court's analysis turned on its reduction of "data field type" to "data type," and on its further holding that "checking a value is not determining a … data field type." JA50; JA134-37. However, Realtime again offered substantial evidence creating, at the very least, a genuine dispute of material fact as to whether—in determining a

data field type—the accused products categorize the data in the field by data type. JA1760. The court erred in concluding otherwise.[13]

The claims at issue include analysis of the data field's content to determine or recognize a data field type. JA133. The data field type is in turn associated with an optimal encoding technique for that field. JA430(23:38-41). Realtime's evidence demonstrated that all of the accused products analyze the content of each data field to categorize the data in that field as one of three possible data types: a "redundant" data type; a "sequentially different" data type; or a "most common value" data type. JA1751-53; JA1760; JA529-30. These three data types are associated in turn with three optimal encoding techniques: "redundant" with copy encoding; "sequentially different" with increment encoding; and "most common value" with default encoding.[14] JA1751-53; JA1760; JA529-30.

With respect to copy encoding, the accused products compare the content of the data field in the current message of a data packet with the content of the corresponding data field in a prior message of the same data packet. If the

[13] The court's summary judgment orders on this ground deal with two sets of claim limitations—the so-called "analyzing" and "selecting" limitations. JA133-39; JA141-42. Because the court's analysis for both of these sets of limitations is the same, JA133-39; JA141-42, Realtime addresses them together in this brief.

[14] All of the accused products use [redacted]. JA1759; JA4224-27. [redacted] are used by all of the accused products except for those products identified in Dr. Shamos' declaration and infringement charts. JA1762; JA1773; JA3767-JA4082.

relationship between the content of the two fields is such that they are the same, the field in the current message is of the data type “redundant,” and copy encoding is selected as the optimal technique. JA1752. Similarly, with respect to increment encoding, the accused products compare the content of the data field in the current message with the content of the corresponding data field in a prior message. If the relationship between the content of the two fields is such that they have a difference of one, the field in the current message is of the data type “sequentially different,” and increment encoding is selected as the optimal technique. JA1752. And with respect to default encoding, the accused products compare the content of the data field in the current message with an initial value defined by the template. If the content of the data field is equivalent to this *a priori* value, the field is of the data type “most common value,” and default encoding is selected as the optimal technique. JA1751-53; JA1760; JA529-30.

Dr. Shamos confirmed that this functionality meets the district court’s claim construction by “categoriz[ing] the data in the field (or block) as one of … other data type”—that is, as one of “redundant,” “sequentially different,” or “most common value” data type. JA1751-53; JA1760. Indeed, these are characteristic logical types, and logical types are a commonly recognized example of “other” data types. JA5057-58; JA5072-74 (“data type” includes “logical type”). Dr. Shamos further confirmed that this functionality is not fairly described as simply

"checking a value"—the products do not, for example, simply check for the number "212." Instead they categorize the data by comparing the value in one field with the value in another (or in a template) and determining the data field type based on the *logical relationship* between those values. JA1752; JA539.

Again, the district court should have credited Realtime's evidence, and drawn inferences in Realtime's favor. *Absolute Software*, 659 F.3d at 1130. But it did not do so. Instead it credited the factual opinion of Defendants' expert: "[t]his is, as Dr. Storer states …, really a value check—not a content categorization." JA136. The court suggested that this was "uncontradicted by Dr. Shamos." JA136. But Dr. Shamos *did* contradict the opinion that determining the data field type involves only a so-called "value check," and not a content categorization.[15] JA1751-53; JA1760. The district court also pointed to Dr. Storer's opinion that the accused products do not "determin[e] whether the content is ASCII, multimedia, signed or unsigned integers, etc." JA136. That may be true as far as it goes, but Dr. Shamos explained that the products *do* determine whether the content is of "other

[15] As noted, Dr. Storer offered no *Markman*-related testimony regarding "checking a value." JA793-98. At the summary judgment stage, however, he offered an opinion that the accused products fail the "value check" limitation that the district court imported into its claim construction. JA5181. As noted, that opinion was contradicted and undermined at the summary judgment stage by Dr. Shamos. JA1751-53; JA1760. Dr. Storer's opinion was further undermined by the conflicting opinion of Dr. Brogioli—Defendants' expert on invalidity—who took the position that the relevant claim limitations are met when "encoders are selected based on analyzing the value of a data field." JA4228; JA5202.

data type"—a phrase the court included in its construction to permit "an expansive reading of this term." JA1760; JA48. The promise of that expansive reading at claim construction was not fulfilled at summary judgment.

Even under the court's construction of "data field/block type," Realtime's summary judgment evidence raised, at the very least, a genuine factual dispute as to whether the accused products categorize the data in the field as a "redundant," "sequentially different," or "most common value" data type. JA1751-53; JA1760; JA529-30. The district court's holding to the contrary should be reversed.

3. There is a genuine dispute regarding whether the accused products meet the "data stream" limitation.

The district court also granted summary judgment of noninfringement on the four asserted claims of the '568 patent as to ISE, NYSE, and OPRA, and on fifteen asserted claims of the '747 and '651 patents as to ISE alone, for an alleged failure to meet the "data stream" limitation under the court's construction of that term as "one or more blocks transmitted in sequence from an external source whose characteristics are not controlled by the data encoder or decoder." *See* Issues Chart. The court's analysis turned on the imported "external source" and "not controlled" limitations. JA124-28. Once again, however, Realtime offered evidence creating, at the very least, a genuine dispute of material fact as to whether the encoders in the accused products operate on a data stream satisfying these limitations. JA1754-55; JA536; JA1811. The district court erred in concluding otherwise.

As an initial matter, Realtime's evidence demonstrated that the accused encoders operate on just the kind of data stream described in the specifications. FAST documentation confirmed that the "FAST Protocol is an encoding algorithm which reduces the size of a data stream," and further that the "FAST *stream* consists of a sequence of messages or a sequence of blocks." JA536; JA1811.

Dr. Shamos also demonstrated, in detailed product-by-product infringement charts, that each of the accused ISE, NYSE, and OPRA data encoders operate on a "financial data stream." JA2734-861; JA3767-4082. Dr. Shamos explained that, for all of these encoders, one or more data blocks are transmitted in sequence from an external source that generates the market data—such as the market data server or ticker plant generating FIX messages—to the data encoder, which receives and encodes that data stream. JA1754. He noted that this conclusion was supported by his review of the source code for these FAST encoders, which confirmed that the encoders "cannot and do not generate [redacted]." JA1754; JA4300(95:24-96:2) ("[The data stream is] not coming from the code that's performing encoding and decoding. It's being supplied to it."). Dr. Shamos further explained that the encoders in the accused products do not control the characteristics of the data blocks or the order in which those blocks are transmitted and compressed. JA1754-55.

This evidence again should have been credited by the district court as raising a genuine dispute of material fact, and summary judgment should have been denied on that basis. *Absolute Software*, 659 F.3d at 1130. The court offered three reasons for its contrary conclusion. JA124-28; JA154-57; JA162. None has merit.

First, with respect to NYSE and OPRA, the district court found "an absence of any specific facts relating to how any of the specific encoding products work." JA156. The court complained that "nothing in Dr. Shamos' declaration … specifically addresses the NYSE and OPRA data streams." JA157. But Dr. Shamos' declaration—which expressly covered the accused NYSE and OPRA encoding products—was accompanied by hundreds of pages of infringement charts specifically addressing these products, along with the source code supporting the declaration and the charts. JA3767-4082. Those charts covered each of the systems highlighted in the court's summary judgment order: OPRA Encoding, ArcaBook Encoding, Filtered Options Feed Encoding, and XDP Depth of Book Encoding.[16] JA3971-4044; JA4057-4082. The court simply failed to account for this substantial and specific infringement evidence regarding the products at issue.

Second, the court concluded that the NYSE, OPRA, and ISE encoders do not operate on a data stream "from an external source" because these Defendants

[16] Contrary to the district court's suggestion, JA154, Realtime never abandoned its claim with respect to XDP Depth of Book. JA4024-44; JA5154.

generate—somewhere in their system—the financial messages that are ultimately encoded prior to transmission. The court found, for example, that "ISE generates its feed internally from stored data. … 'ISE feeds' are internal (and therefore not external) to ISE." JA124. Likewise, the "messages that [NYSE] compresses are generated by the back-end server itself; the messages therefore do not come from an external source." JA156; JA162 ("NYSE's and OPRA's accused encoding instrumentalities do not meet the 'data stream' claim element … because [they] encode data that comes from within NYSE's and OPRA's own systems."). The court applied a similar analysis to conclude that certain ISE encoders do not output a "data stream." JA127 ("since the [ISE] journal records are never sent to a third party they are not part of any output data stream").

The court's error here was applying the imported "external source" limitation to require a source *external to the Defendant*—that is, a third party—rather than a source *external to the encoder*. The court's construction cannot reasonably require that a third party generate the encoded data stream. Indeed, such a requirement conflicts with the patent specifications, which expressly envision that financial institutions such as NYSE will use the disclosed inventions to encode the financial messages they generate and transmit them to customers such as traders. JA413-14(Figs.1&2); JA422(8:35-52). Under the court's application of its "external source" limitation, the claims preclude the possibility that financial

institutions will encode their own messages prior to transmission to their customers. JA124-28; JA162. That excludes the preferred embodiments and—even more fundamentally—makes little sense. *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583 (Fed. Cir. 1996) (noting that claim interpretations excluding preferred embodiments are "rarely, if ever, correct").

Third, the court found that the NYSE and OPRA systems fail to meet the "not controlled" limitation because "the data that enters those instrumentalities is internally altered in some fashion and out of the original sequence." JA157. That is, the court credited Defendants' arguments that characteristics of the data stream are controlled by certain software in their systems including the "PartiApps," "HslApps," "Front-End Server," and "FOF server." JA154-57; JA4274-75.

The court's error here was analogous to its error with respect to "external source"—while the construction precludes control of data stream characteristics "by the data *encoder or decoder*," the court applied it to preclude control of data stream characteristics anywhere in Defendants' systems. JA63(emphasis added); JA154-57. The PartiApps, HslApps, Front End Server, and FOF server are larger applications or servers performing numerous tasks, and they include many functions *external* to the data encoders or decoders in the accused systems. JA1754-55; *see, e.g.*, JA2798-99; JA3893; JA4062-63. As Dr. Shamos testified, the "requirement that's listed here in the [district court's] construction is that the

data encoder or decoder can't control the characteristics of the source, and they don't … for any of the defendants. The data encoder and decoder don't control the characteristics of the source. They have to live with the characteristics of the source." JA4300(96:10-18) (emphasis added).

Even under the court's construction of "data stream," Realtime's summary judgment evidence raised, at the very least, a genuine factual dispute as to whether the accused encoders operate on "one or more blocks transmitted in sequence from an external source whose characteristics are not controlled by the data encoder or decoder." JA1754-55; JA1811. The court's contrary holding should be reversed.

C. The District Court Erred in Granting Summary Judgment of Noninfringement on Claims 95, 97, 108, and 112 of the '651 Patent By Applying an Agreed Construction to the Claims in a Manner That Excludes the Preferred Embodiment.

In addition to granting summary judgment of noninfringement on the basis of its own constructions, the district court granted summary judgment as to four asserted "decoding" claims on the basis of an agreed construction. *See* Issues Chart. In so doing, the court applied this construction—that "the system (or method) selects the lossless encoders based on analyses of content of the data blocks (or data fields)"—to import an extraneous *encoding* requirement into the language quoted below of four *decoding* claims of the '651 patent.[17] JA130-31;

[17] The claim term construed was "wherein the lossless encoders are selected based on analyses of content of the data blocks [or fields]." JA1406.

JA1406. That is, the court interpreted the "system (or method)" in the agreed construction as referring to the *decoding* system or method in the preamble, and not the *encoding* system or method that generated the received data packet. JA130. This limitation makes little sense in the context of the claim language. Indeed, it is clear from a plain reading of the claim language itself, quoted below, that the encoding language in the "decoding" claims merely describes the characteristics of the encoded data packet so it can be decoded. JA523-24.

1. The agreed construction did not import an encoding requirement into the asserted decoding claims.

The '651 patent contains both encoding and decoding claims. JA519-24. The specification makes clear that encoding and decoding are complimentary processes; the claimed decoding systems and methods reverse the compression achieved by the corresponding encoding systems and methods. JA514(13:35-14:44). But the required symmetry between encoding and decoding does not mean that the '651 patent teaches one invention that necessarily requires both encoding and decoding, or necessarily requires one party or system to perform both functions. In fact the specification envisions that one party may generate and encode the data, while another receives and decodes that encoded data. JA503(Fig.2); JA512(9:35-36,9:53-56); JA511(8:46-9:3). Indeed, this is the purpose of the disclosed "descriptors," which ensure that the symmetry between encoding and decoding is maintained without any need for the decoding party to

have participated in the encoding process. JA511-12(8:46-9:36); JA515(16:18-23); JA516(17:18-22); JA1490-92.

The asserted decoding claims, much like those addressed by this Court in *Uniloc* and *SiRF*, are structured "to capture infringement by a single party"—that is, they focus exclusively on decoding, "and define[] the environment in which that [decoding] must function." *Uniloc USA, Inc. v. Microsoft Corp.*, 632 F.3d 1292, 1309 (Fed. Cir. 2011); *see also SiRF Tech., Inc. v. ITC*, 601 F.3d 1319, 1329 (Fed. Cir. 2010). Claims 95 and 97, for example, recite a "method of decoding" that includes the step of:

> **receiving an encoded data packet** from the financial data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders **used** to encode data blocks associated with the encoded data packet, and further wherein the lossless encoders are **selected** based on analyses of content of the data blocks.

JA523 (emphasis added). Claims 108 and 112 recite similar language. JA524. This language makes clear that the decoding systems or methods do not themselves encode data—they receive data packets that have already been encoded. The subordinate "having," "wherein," and "further wherein" clauses simply describe how the encoded data was encoded (*i.e.*, what encoders were "used" and how they were "selected" by the encoding system or method). *See Uniloc*, 632 F.3d at 1309.

Nothing in the agreed construction changes this analysis. As noted, that construction—proposed by Defendants—swapped the further "wherein" clause

quoted above for "the system (or method) selects the lossless encoders based on analyses of content of the data blocks (or data fields)." JA1406; JA523-24(claims 91, 108). This construction did not change the meaning of the "wherein" clause, which continues to describe "the environment in which [the claimed decoding] must function." *Uniloc*, 632 F.3d at 1309. Defendants argued in summary judgment—and the court agreed—that "the system (or method) [that] selects the … *encoders*" in the agreed construction refers to the claimed *decoding* system (or method). But it plainly refers to the *encoding* system that previously encoded the data now being received and operated upon by the decoding system. JA1406; JA523-24(claims 91, 108). Indeed, only encoding systems would "select[] lossless encoders;" the claimed decoding systems, on the other hand, expressly "select one or more lossless decoders." *Id.*

The district court, however, again credited Defendants' arguments. "According to [Defendants]," the court noted, "Dr. Shamos concedes that these decoding claims require encoding." JA130. But Dr. Shamos conceded no such thing; in fact he submitted a declaration explaining why no encoding requirement had been imported into these decoding claims. JA1487-92. The court nevertheless concluded that "to infringe these [decoding] claims a single party … must itself perform … both the encoding and decoding steps." JA130. The court then granted summary judgment of *noninfringement* because the evidence indicated that—just

as in the preferred embodiment—the decoding systems accused of infringing these decoding claims often "had nothing to do with the original encoding." JA131. Thus, according to the court, they did "not perform an essential [encoding] step of the [decoding] claim[s]." JA131. This holding improperly narrowed the agreed construction, read the preferred embodiment out of the claims, and ignored the declaration of Dr. Shamos explaining that the decoding claims contain no encoding step.[18] JA1492; *Vitronics Corp.*, 90 F.3d at 1583; *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1343-44 (Fed. Cir. 2001). The summary judgment order on these decoding claims should be reversed.

2. This Court should consider providing a clarifying construction for trial on remand.

While Realtime agreed to the reformulation of the "wherein" clause quoted above, the disputes that subsequently arose over that construction demonstrate that its "system (or method)" reference contains an unintended latent ambiguity as to whether the construction is pointing to the encoding system or the decoding system. In circumstances such as these, this Court may provide a clarifying construction for trial on remand. *Decisioning.com, Inc. v. Federated Dep't Stores,*

---

[18] The district court suggested that Realtime offered an argument that the decoding Defendants "mastermind[ed]" the original encoding, JA131, but in fact Realtime offered a "mastermind" argument in relation to different claims from a different patent. With respect to claims 95, 97, 108, and 112 of the '651 patent, Realtime argued that Defendants were wrong to suggest that the agreed construction imported an encoding limitation into these decoding claims. JA1487-92.

Inc., 527 F.3d 1300, 1312-13 (Fed. Cir. 2008); *see also Meyer Intellectual Props. Ltd. v. Bodum Inc.*, 690 F.3d 1354, 1368-69 (Fed. Cir. 2012). Should the Court do so here, Realtime respectfully proposes that the agreed construction be modified as follows: "the lossless encoders are selected by the encoding system (or method) based on analyses of content of the data blocks (or data fields)."

D. The District Court Erred in Granting Summary Judgment of Invalidity on Claims 1, 4, 6, 7, and 12 of the '651 Patent and Claims 1, 7, 8, and 13 of the '747 Patent.

While the district court granted summary judgment of noninfringement on four of the asserted decoding claims, it granted summary judgment of invalidity on nine others. *See* Issues Chart. Claim 1 of the '747 patent contains the representative limitations that the court determined to be invalidating:

> decompressing the data block with a selected lossless decoder utilizing **content dependent data decompression**, if the descriptor indicates the data block is encoded utilizing **content dependent data compression**; and
>
> decompressing the data block with a selected lossless decoder utilizing **content independent data decompression**, if the descriptor indicates the data block is encoded utilizing **content independent data compression**.

JA491(26:35-43) (emphasis added); JA69-70. The court found the highlighted "compression" terms definite and supported by the written description. JA70-72. It found the corresponding "decompression" terms, on the other hand, indefinite and lacking written description support. JA84-85. This conclusion, however, conflicts

with the evidence—including evidence from both sides. The summary judgment order on these decoding claims should also be reversed.

1. These claims comply with the definiteness requirement.

A claim is definite so long as "one skilled in the art would understand the bounds of the claim when read in light of the specification." *Exxon Research & Eng'g Co. v. United States*, 265 F.3d 1371, 1375 (Fed. Cir. 2001). Reasonable disagreements and manufactured confusion are no threat to validity: "[i]f the meaning of the claim is discernible," then "close questions of indefiniteness … are properly resolved in favor of the patentee." *Id.* at 1375, 1380; *Bancorp Servs., L.L.C. v. Hartford Life Ins. Co.*, 359 F.3d 1367, 1372 (Fed. Cir. 2004); *Ultimax Cement Mfg. Corp. v. CTS Cement Mfg. Corp.*, 587 F.3d 1339, 1352 (Fed. Cir. 2009) (holding that a term is definite if it "can be given any reasonable meaning").

The district court had no difficulty understanding and construing the compression terms. As the court found, the patents describe compression methods "in which content dependent data compression ['CDDC'] is performed on a data block when the data type … is identified; when the data type is not identified, then content independent data compression ['CIDC'] is utilized." JA71. Dr. Shamos explained—as one of skill in the art himself—that the corresponding decompression terms are understood by persons skilled in the art as describing the methods to reverse CDDC and CIDC. JA900-01; JA1201-11; JA5078.

This straightforward understanding of the CDDD and CIDD terms is supported by the claims themselves, which indicate that CDDD is used by the decoder if the descriptor indicates that CDDC was used by the encoder, and likewise that CIDD is used by the decoder if the descriptor indicates that CIDC was used by the encoder. JA491(26:35-43); JA69-70. This understanding is also supported by the specifications, which explain that the "decoders correspond[] to the extracted compression type descriptor." JA516(17:31-32); JA460(Fig.12). That is, again, CDDD will correspond to CDDC, and CIDD will correspond to CIDC.

This straightforward understanding is also supported by Defendants' own submissions regarding the alleged anticipation of these claims. With respect to claim 1 of the '747 patent, for example, Defendants argued that

> the XDemill decompressor parses the structure container to identify the corresponding decoders to use for decompression. If a content-dependent semantic compressor was used to compress the data, a **corresponding content-dependent decompressor** will be used to decompress.

JA1218-19 (emphasis added). Defendants made similar arguments regarding other references. JA1220-21; JA1229; JA1232 ("Sebastian discloses decoding the data field … utilizing content independent data decompression, if the descriptor indicates the data field is encoded utilizing content independent data compression."). In short, Defendants understood and applied the CDDD and CIDD

terms just as Dr. Shamos explained that anyone of skill in the art would—until it came time to make an indefiniteness argument.[19]

The district court nevertheless concluded, again crediting Defendants' arguments, that these decompression terms "have no meaning." JA69. The court's reasoning went as follows: 1) decompression is always the reverse of compression; 2) decompression is never based on an analysis of compressed content—indeed, analysis of content must necessarily come before that content is compressed; 3) the decompression terms CDDD and CIDD are meaningless because they should be read to require a nonsensical analysis of compressed content, rather than to signify the reverse of CDDC and CIDC. JA75-85.

This reasoning turns the law of indefiniteness on its head. Rather than asking whether CDDD and CIDD could "be given any reasonable meaning," *Ultimax*, 587 F.3d at 1352, the court instead sought—and found—a reading of these terms that rendered them meaningless. The cited declarations and intrinsic record demonstrate that the terms can be given a reasonable meaning: CDDD is a process that reverses CDDC, and CIDD is a process that reverses CIDC. JA900-01;

[19] These arguments were originally made by certain Defendants in *inter partes* reexamination proceedings involving the '747 and '651 patents. JA1212-33. They were subsequently adopted by all Defendants in the invalidity contentions served on Realtime in this litigation.

JA1218-21; JA1229-33; *Bancorp*, 359 F.3d at 1375-76 (noting that defendants' use of the term confirmed a discernible meaning).

2. These claims comply with the written description requirement.

The district court recognized that Defendants' indefiniteness and written description arguments stood or fell together. JA77 ("All of defendants' arguments turn on the same fundamental proposition … that no part of the data decompression process requires … reference to the data content."). If, as Dr. Shamos explained, CDDD and CIDD are understood to signify processes that reverse the effects of CDDC and CIDC, then the specifications contain ample written description to support these decompression terms. JA1201-1211; JA479(1:19-22) ("The present invention … us[es] content independent and content dependent data … decompression."); JA481(5:45-51); JA459-60(Figs.11&12); JA508(1:19-30); JA510(5:66-67); JA514(13:57). If, on the other hand, CDDD and CIDD are understood to signify a nonsensical analysis of compressed content, then the specifications do not support the terms. Because, as demonstrated above, the decompression terms should be given their sensible and straightforward meaning, these claims comply with the written description requirement. The district court erred in concluding otherwise. JA85.

The court also erred in concluding that "content dependency and independency … lose all meaning once the encoding process has occurred and the

descriptor is appended." JA84. The court apparently believed there was no support for the descriptor to indicate *both* the specific encoder and its class (content dependent or independent), as recited in the claims.[20] JA491(26:26-27,35-42). But that is simply not true. JA486(15:31-37) ("the *data compression type descriptor* may mandate the application of: a single specific decoder, … a class or family of decoders, … or any combination or permutation thereof") (emphasis added); JA485(14:54-59) ("The data compression type descriptor may possess values … in accordance with the data compression system embodiments"); JA486(15:50-60).

In support of its erroneous written description conclusion, the court pointed to the deposition of an attorney involved in drafting applications in the chain leading to the '747 patent. JA83. But that attorney conceded that he "wasn't an expert at decoding," JA1351(91:6), and this Court has held that a witness may not "testify as an expert on … invalidity unless that witness is qualified as an expert in the pertinent art." *Sundance, Inc. v. Demonte Fabricating Ltd.*, 550 F.3d 1356, 1363 (Fed. Cir. 2008). Dr. Shamos, on the other hand, is qualified as an expert in the pertinent art—and he testified that the relevant specifications disclose that Realtime's inventors were in possession of the claimed decompression inventions

[20] The court apparently decided that the latter classification was "superfluous," and hence indefinite, in the context of decoding. JA83. But whether that limitation was necessary is not the proper test for indefiniteness, and it was error for the court to so conclude.

in the relevant timeframe. JA1209-11; JA1261-1311. This testimony was again sufficient to create, at the very least, a genuine dispute of material fact as to whether these decompression terms complied with the written description requirement. *See Laryngeal Mask Co. Ltd. v. Ambu A/S*, 618 F.3d 1367, 1373 (Fed. Cir. 2010) ("[patents must] convey with reasonable clarity to those skilled in the art that … [the patentees were] in possession of the [claimed] invention"). The district court again erred in concluding otherwise. JA85.

E. The District Court Abused Its Discretion in Precluding Realtime From Asserting Infringement Under the Doctrine of Equivalents.

There is no dispute that Realtime timely disclosed its intent to assert infringement under the doctrine of equivalents. In July and October 2010, shortly after these actions were filed in Texas, Realtime served infringement contentions disclosing that, "[t]o the extent that any elements of the asserted claims are not deemed to be literally present, Realtime … asserts that those elements are present in the Accused Instrumentalities or their use under the doctrine of equivalents." JA4458-61; JA4490; JA4553. These disclosures complied with the local rules applicable in Texas and in New York, where these actions were later transferred, which require that Realtime state "[w]hether each limitation of each asserted claim is alleged to be literally present or present under the [DOE] in the Accused Instrumentality." JA94; *see also* E.D. TEX. PATENT R. 3-1(d); JA89.

On June 15, 2012, Realtime served an expert report on infringement from Dr. Shamos which, reflecting those 2010 disclosures, offered opinions "as to infringement under the doctrine of equivalents." JA95. The report was timely as well—it analyzed infringement under Defendants' proposed claim constructions, and was served a week before the district court issued its *Markman* opinion. JA93-94. Furthermore, nothing in the report prejudiced Defendants. They had long been on notice that Realtime intended to assert infringement under the DOE, had never moved for more detailed DOE contentions, and their own experts had no trouble responding to each of Dr. Shamos' DOE-related opinions in their rebuttal reports. JA4624-25(50:8-11) ("MS. HUTTNER: … We have experts that have addressed the doctrine of equivalents."); JA4458-61; JA4742-43.

The district court nevertheless granted Defendants' motion to strike Dr. Shamos' DOE-related testimony, and entered its final judgments precluding Realtime from asserting infringement under the doctrine of equivalents. JA87-99; JA1-24. That was an abuse of discretion. *Cybor*, 138 F.3d at 1460.

The court noted that Realtime filed supplemental infringement contentions in March and April of 2012 that did not address the DOE. JA91. Those contentions, however, were *supplemental*—they supplemented, but did not supersede, the original contentions asserting infringement under the DOE. JA4459-

60. The court thus clearly erred in concluding that Realtime "failed to preserve infringement claims under the doctrine of equivalents." JA94.

The district court also clearly erred in finding that Defendants were prejudiced by Dr. Shamos' DOE-related testimony. JA96-99. The court noted that its *Markman* opinion had "construed the claims consistently with defendants' proposed constructions." JA94. And the court was troubled by what it termed "a sort of 'heads I win, tails you lose' scenario, [in which] the Shamos Report finds infringement under either plaintiff's or defendants' proposed constructions." JA95. This troubled the court because, it suggested, Defendants would likely have taken different *Markman* positions had they "understood that plaintiff would assert that different claim constructions would result in 'insubstantial differences' regarding infringement." JA98-99. This analysis of the alleged prejudice to Defendants again reflected an erroneous view of the *Markman* process as focused on "the objective of capturing or excluding the accused device[s]." *Vita-Mix*, 581 F.3d at 1324.

Defendants' "*Markman* positions"—particularly in light of their adoption by the district court—should have been based on the intrinsic evidence, not on considerations "regarding infringement." JA98-99. The court's apparent assumption that its adoption of Defendants' proposed constructions left no room for any finding of infringement was thus clear error, and the order precluding Realtime from asserting infringement under the DOE should be reversed. JA99.

VIII. CONCLUSION AND RELIEF REQUESTED

For all of these reasons, this Court should: reverse the district court's construction of the terms "descriptor indicates," "data field/block type," and "data stream" and adopt Realtime's proposed constructions of those terms; reverse the summary judgments of noninfringement based on those terms; reverse the additional summary judgments of noninfringement on claims 95, 97, 108, and 112 of the '651 patent; reverse the summary judgments of invalidity on the relevant claims of the '747 and '651 patents; and reverse the order precluding Realtime from asserting infringement under the doctrine of equivalents. This Court should then vacate the final judgments and remand this litigation to the district court.

SUMMARY OF DISTRICT COURT'S ERRONEOUS RULINGS AGAINST REALTIME—THE ISSUES CHART

Asserted Claim	Misconstrued Claim Term			Precluded Assertion of DOE	SJ of Invalidity Granted—§112 ¶¶ 1,2	SJ of Noninfringement Granted—Failure to Meet Claim Term			
	"descriptor indicates"	"data field/ block type"	"data stream"			"descriptor indicates"	"data field/ block type"**	"data stream"	"wherein … encoders are selected"
'568 - 15	X	X	X	X		1	1	2	
20		X	X	X			E	2	
22		X	X	X			E	2	
32	X	X	X	X		1	1	2	
'747 - 1	X	X††	X	X	A				
7	X	X††	X	X	A				
8	X	X††	X	X	A				
13	X	X††	X	X	A				
14	X	X	X	X		E	E	3	
19	X	X	X	X		E	E	3	
'651 - 1	X	X	X	X	A				
4	X	X	X	X	A				
6	X	X	X	X	A				
7	X	X	X	X	A				
12	X	X	X	X	A				
13	X	X	X	X		E	E	3	
18	X	X	X	X		1	1	3	
19	X	X	X	X		1	1	3	
21	X	X	X	X		E	E	3	
22	X	X	X	X		E	E	3	
25	X	X	X	X		1	1	3	
26	X	X	X	X		E	E	3	
29	X	X	X	X		E	E	3	
34	X	X	X	X		1	1	3	
35	X	X	X	X		E	E	3	
43	X	X	X	X		E	E	3	
47	X	X	X	X		1	1	3	
49	X	X	X	X		1	1	3	
95	X	X††	X	X		A	A		4
97	X	X††	X	X		A	A		4
108	X	X††	X	X		A	A		4
112	X	X††	X	X		A	A		4

X The ruling applies to the claim.
A The ruling applies to all Defendants.
E The ruling applies to all Exchange Defendants and Morgan Stanley ("Encoding Defendants").
1 The ruling applies to all Encoding Defendants except BATS.
2 The ruling applies to Defendants ISE, NYSE, and OPRA.
3 The ruling applies to Defendant ISE.
4 The ruling applies to all Defendants except CME and OPRA.
** The summary judgment ruling on "data field/block type" included the "analyzing" and "selecting" limitations, which incorporate the term "data field/block type."
†† The term "data block type" is incorporated in the claim through the term "analyses of content of the data blocks," which was construed as "directly examining the content of the data to be compressed to determine the data block type of that data block." JA1404.

Respectfully submitted,

<u>/s/ Dirk D. Thomas</u>
Dirk D. Thomas
MCKOOL SMITH P.C.
1999 K Street, Suite 600
Washington, DC 20006
(202) 370-8302

Joel L. Thollander
MCKOOL SMITH P.C.
300 W. 6th Street, Suite 1700
Austin, TX 78701
(512) 692-8735

Robert A. Cote
Brett E. Cooper
Daniel J. Melman
Laura Handley
Lauren L. Fornarotto
MCKOOL SMITH P.C.
One Bryant Park, 47th Floor
New York, New York 10036
(212) 402-9400

Attorneys for Plaintiff-Appellant Realtime Data, LLC (doing business as IXO)

ADDENDUM

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, Plaintiff,	§	
v.	§	No. 11 Civ. 6697 (KBF) No. 11 Civ. 6699 (KBF) No. 11 Civ. 6702 (KBF)
CME GROUP INC., et al., Defendants.	§	

FINAL JUDGMENT
PURSUANT TO FED. R. CIV. P. 54(b)

Plaintiff Realtime Data, LLC d/b/a IXO ("Realtime") brought the above-captioned actions (the "Action") against Defendants BATS Trading, Inc. and BATS Exchange, Inc. (collectively, "BATS"); CME Group Inc., Board of Trade of the City of Chicago, Inc. and New York Mercantile Exchange, Inc. (collectively, "CME"); International Securities Exchange ("ISE"); NASDAQ OMX Group, Inc. and NASDAQ OMX PHLX, Inc.[1] (collectively, "NASDAQ"); NYSE Euronext, NYSE Arca, Inc., NYSE Amex, LLC and Securities Industry Automation Corporation (collectively, "NYSE"); and Options Price Reporting Authority ("OPRA") (collectively, the "Defendants"), alleging that the Defendants infringe U.S. Patent Nos. 7,417,568 ("the '568 Patent"), 7,777,651 ("the '651 Patent"), and 7,714,747 ("the '747 Patent").

[1] NASDAQ OMX PHLX, Inc. is now known as NASDAQ OMX PHLX LLC.

On June 22, 2012, the Court issued an Opinion and Order (*Markman*) in this Action (D.I. No. 651 in Case No. 11 Civ. 6697).

On June 27, 2012, the Court issued an Opinion and Order (Partial Motion for Summary Judgment re Data Decompression) (D.I. No. 654 in Case No. 11 Civ. 6697) in this Action finding that claims 1, 7, 8 and 13 of the '747 Patent, and claims 1, 4, 6, 7 and 12 of the '651 Patent are invalid under 35 U.S.C. § 112, first and second paragraphs, for failure to comply with the definiteness and written description requirements.

On September 24, 2012, the Court issued an Opinion & Order (D.I. No. 843 in Case No. 11 Civ. 6697) in this Action granting certain motions for summary judgment of non-infringement filed or joined by Defendants in this Action and the related actions (with which this Action was consolidated for pre-trial proceedings) *Realtime Data, LLC d/b/a IXO v. Morgan Stanley, et al.* (Case Nos. 11 Civ. 6696, 6701, 6704) and *Realtime Data, LLC d/b/a IXO v. Thomson Reuters, et al.* (Case Nos. 11 Civ. 6698, 6700, 6703) (the "Related Actions"), and finding that all of the Defendants' accused infringing instrumentalities fail to meet one or more of the elements of each asserted claim of the '568 Patent, the '747 Patent and the '651 Patent.

On November 9, 2012, the Court issued a Supplemental Order granting certain additional motions for summary judgment of non-infringement filed or joined by Defendants in this Action and the Related Actions.

The Court's Opinions and Orders resolve the claims brought by Realtime in its pleadings against each Defendant in this Action, and there being no just reason for delay, pursuant to Federal Rule of Civil Procedure 54(b), the Court hereby enters Final Judgment of non-infringement and invalidity of the specific patent claims addressed in the subject Opinion and Orders referenced above, including as follows:

- Defendant BATS does not infringe claims 20 and 22 of the '568 Patent, claims 13, 21, 22, 26, 29, 35, 43, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of BATS's accused instrumentalities meets the claim element "analyzing [or analyze] content of the [or a] data block [or field of the message] to determine a data block [or field] type," "analyses of content of [the] data blocks [or fields]," "recognizing a data field type of a data field"[2] or "recognizing data field types;"[3]
- Defendant BATS does not infringe claims 13, 21, 22, 26, 29, 35, 43, 95, 97, 108 and 112 of the '651 Patent and claims 14 and 19 of the '747 Patent because none of BATS's accused instrumentalities meets the claim element "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders," or "select one or more lossless encoders;"
- Defendant BATS does not infringe claims 13, 21, 22, 26, 29, 35, 43, 95, 97, 108 and 112 of the '651 Patent and claims 14 and 19 of the '747 Patent because none of BATS's accused instrumentalities meets the claim element "descriptor with the encoded data which identifies," "descriptors indicate" or "descriptor indicates;"
- Defendant BATS does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for BATS's accused decoding instrumentalities, BATS does not perform

[2] This term was construed in the in the July 9, 2012 Order Adopting the Parties' Agreed Claim Constructions (D.I. No. 667 in Case No. 11 Civ. 6697) ("Agreed Claim Construction Order") as "recognizing a data field type by analyzing the content of the data field."

[3] While "recognizing data field types" was not specifically construed in the Agreed Claim Construction Order, claim 20 of the '568 Patent, which contains that term, was listed in the claim term chart next to "recognizing a data field type of a data field," and the Court hereby adopts as the construction of "recognizing data field types" the same construction that it adopted for the substantially identical term "recognizing a data field type of a data field." (*See* Agreed Claim Construction Order at 2.)

(or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"

- Defendant CME does not infringe claims 15, 20, 22 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of CME's accused instrumentalities meets the claim element "analyzing [or analyze] content of the [or a] data block [or field of the message] to determine a data block [or field] type," "analyses of content of [the] data blocks [or fields]," "recognizing a data field type of a data field" or "recognizing data field types;"
- Defendant CME does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of CME's accused instrumentalities meets the claim element "descriptor with the encoded data which identifies," "descriptors indicate" or "descriptor indicates;"
- Defendant CME does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of CME's accused instrumentalities meets the claim element "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders," or "select one or more lossless encoders;"
- Defendant ISE does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108, and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent, because none of ISE's accused

instrumentalities meets the claim element "descriptor with the encoded data which identifies," "descriptors indicate" or "descriptor indicates;"

- Defendant ISE does not infringe claims 15, 20, 22 and 32 of the '568 Patent because none of ISE's accused encoding instrumentalities meets the claim element "data stream;"
- Defendant ISE does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of ISE's accused instrumentalities meet the claim element "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders," or "select one or more lossless encoders;"
- Defendant ISE's accused instrumentality ISE.FastProcessing Encoding does not infringe claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47 and 49 of the '651 Patent and claims 14 and 19 of the '747 Patent because that instrumentality does not meet the claim element "data stream;"
- Defendant ISE does not infringe claims 15 and 32 of the '568 Patent because none of ISE's accused encoding instrumentalities meets the claim element "packet" or "data packet;"
- Defendant ISE's accused instrumentalities Exegy OPRA Decoding, OPRA Decoding and ISE.FASTSpec Greeks Decoding do not infringe claims 95, 97, 108 and 112 of the '651 Patent because for those accused decoding instrumentalities, ISE does not perform (or directly control another who performs) the claim

limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"

- Defendant ISE's accused instrumentality MIDAS Encoding does not infringe claims 20 and 22 of the '568 Patent because that instrumentality does not meet the claim elements "processing the description file with a data compression compiler" and "outputting an executable file that is used to process a stream of data by recognizing data field types in the data stream and applying encoders associated with the recognized data field types to encode the data stream;"
- Defendant ISE does not infringe claims 15, 20, 22 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of ISE's accused instrumentalities meets the claim element "analyzing [or analyze] content of the [or a] data block [or field of the message] to determine a data block [or field] type," "analyses of content of [the] data blocks [or fields]," "recognizing a data field type of a data field" or "recognizing data field types;"
- Defendant NASDAQ does not infringe claims 15, 20, 22 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of NASDAQ's accused instrumentalities meets the claim element "analyzing [or analyze] content of the [or a] data block [or field of the message] to determine a data block [or field] type," "analyses of content of [the] data blocks [or fields]," "recognizing a data field type of a data field" or "recognizing data field types;"

- Defendant NASDAQ does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of NASDAQ's accused instrumentalities meets the claim element "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders," or "select one or more lossless encoders;"
- Defendant NASDAQ does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of NASDAQ's accused instrumentalities meets the claim element "descriptor with the encoded data which identifies," "descriptors indicate" or "descriptor indicates;"
- Defendant NASDAQ does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for NASDAQ's accused decoding instrumentalities, NASDAQ does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"
- Defendants NYSE and OPRA do not infringe claims 15, 20, 22 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of NYSE's and OPRA's accused instrumentalities meets the claim element "analyzing [or analyze] content of the [or a] data block [or field of the message] to determine a data block [or field] type," "analyses of content of [the] data blocks

[or fields]," "recognizing a data field type of a data field" or "recognizing data field types;"

- Defendants NYSE and OPRA do not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of NYSE's and OPRA's accused instrumentalities meets the claim element "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders," or "select one or more lossless encoders;"
- Defendants NYSE and OPRA do not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of NYSE's and OPRA's accused instrumentalities meets the claim element "descriptor with the encoded data which identifies," "descriptors indicate" or "descriptor indicates;" and
- Defendants NYSE and OPRA do not infringe claims 15, 20, 22 and 32 of the '568 Patent because none of NYSE's and OPRA's accused encoding instrumentalities meets the claim element "data stream;"
- Defendant NYSE's accused decoding instrumentalities CEF Ultra Plus Decoding, Eurex EBS Decoding, ISE Depth of Market Decoding, NSX Depth of Book Decoding. Superfeed BATS Decoding, Superfeed BATS Europe Decoding, Superfeed CME Market Depth Decoding, Superfeed SWX MDI Decoding and Xetra EBS Decoding do not infringe claims 95, 97, 108 and 112 of the '651 Patent because for those accused decoding instrumentalities, NYSE does not

perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;" and

- Claims 1, 7, 8 and 13 of the '747 Patent and claims 1, 4, 6, 7 and 12 of the '651 Patent are invalid under 35 U.S.C. § 112, first and second paragraphs, for failure to comply with the definiteness and written description requirements.

Further, the Court hereby enters Final Judgment that Realtime is precluded from asserting infringement under the doctrine of equivalents through its expert or any related testimony. (August 2, 2012 Memorandum and Order (D.I. No. 750 in Case No. 11 Civ. 6697).)

Further, since the effect of this Final Judgment resolves liability as to the Defendants' accused infringing instrumentalities, the Court hereby dismisses as moot and without prejudice any remaining, unresolved counter-claims and affirmative defenses raised by any party in this Action.

Further, it is Ordered that, to the extent not granted or denied by an existing Court Order, all currently pending motions and joinders thereto are denied without prejudice, with leave to renew if an appropriate time arises.

Finally, notwithstanding the above, this Court retains jurisdiction to hear any attorneys' fees motions brought by the Defendants in this Action. The Court will set a briefing schedule for those motions in due course. Accordingly, in the interim, the parties' time to file such motions under Federal Rule of Civil Procedure 54(d)(2)(B) is hereby extended to such date as the Court may set in the future.

SO ORDERED:

Dated: New York, New York
November 9, 2012

K. B. Forrest

KATHERINE B. FORREST
UNITED STATES DISTRICT JUDGE

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, Plaintiff, v. MORGAN STANLEY, et al., Defendants.	§ § § § § § § § § §	No. 11 Civ. 6696 (KBF) No. 11 Civ. 6701 (KBF) No. 11 Civ. 6704 (KBF)

FINAL JUDGMENT
PURSUANT TO FED. R. CIV. P. 54(b)

Plaintiff Realtime Data, LLC d/b/a IXO ("Realtime") brought the above-captioned Actions (the "Action") against Defendants Credit Suisse Holdings (USA), Inc. and Credit Suisse Securities (USA) LLC (collectively, "Credit Suisse"); The Goldman Sachs Group, Inc., Goldman Sachs & Co. and Goldman Sachs Execution & Clearing, L.P. (collectively, "Goldman Sachs"); HSBC Bank USA, N.A. and HSBC Securities (USA), Inc. (collectively, "HSBC"); J.P. Morgan Chase & Co., J.P. Morgan Securities Inc. and J.P. Morgan Clearing Corp. f/k/a Bear, Stearns Securities Corp. (collectively, "J.P. Morgan"); and Morgan Stanley and Morgan Stanley & Co., Inc. (collectively, "Morgan Stanley") (collectively, the "Bank Defendants"), alleging that the Bank Defendants infringe U.S. Patent Nos. 7,417,568 (the "'568 Patent"), 7,777,651 (the "'651 Patent"), and 7,714,747 (the "'747 Patent").

On June 22, 2012, the Court issued an Opinion and Order (*Markman*) in this Action (D.I. No. 498 in Case No. 11 Civ. 6696).

On June 27, 2012, the Court issued an Opinion and Order (Partial Motion for Summary Judgment re Data Decompression) (D.I. No. 501 in Case No. 11 Civ. 6696) in this Action finding that claims 1, 7, 8 and 13 of the '747 Patent, and claims 1, 4, 6, 7 and 12 of the '651 Patent are invalid under 35 U.S.C. § 112, first and second paragraphs, for failure to comply with the definiteness and written description requirements.

On September 24, 2012, the Court issued an Opinion and Order (D.I. No. 575 in Case No. 11 Civ. 6696) in this Action granting certain motions for summary judgment of non-infringement filed or joined by Defendants in this Action and the related actions (with which this Action was consolidated for pre-trial proceedings) *Realtime Data, LLC d/b/a IXO v. CME Group Inc., et al.* (Case Nos. 11 Civ. 6697, 6699, 6702) and *Realtime Data, LLC d/b/a IXO v. Thomson Reuters, et al.* (Case Nos. 11 Civ. 6698, 6700, 6703) (the "Related Actions"), and finding that all of the Bank Defendants' accused infringing instrumentalities fail to meet one or more of the elements of each asserted claim of the '568 Patent, the '747 Patent, and the '651 Patent.

On November 9, 2012, the Court issued a Supplemental Order granting certain additional motions for summary judgment of non-infringement filed or joined by Defendants in this Action and the Related Actions.

The Court's Opinions and Orders resolve the claims brought by Realtime in its pleadings against each Bank Defendant in this Action, and there being no just reason for delay, pursuant to Federal Rule of Civil Procedure 54(b), the Court hereby enters Final Judgment of non-infringement and invalidity of the specific patent claims addressed in the subject Opinion and Orders referenced above, including as follows:

- Bank Defendant Credit Suisse does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Credit Suisse's accused instrumentalities meets the claim element "analyses of content of the data blocks;"
- Bank Defendant Credit Suisse does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Credit Suisse's accused instrumentalities meets the claim element "the lossless encoders are selected;"
- Bank Defendant Credit Suisse does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Credit Suisse's accused instrumentalities meets the claim element "descriptors indicate;"
- Bank Defendant Credit Suisse does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for Credit Suisse's accused instrumentalities, Credit Suisse does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"
- Bank Defendant Goldman Sachs does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Goldman Sachs's accused instrumentalities meets the claim element "analyses of content of the data blocks;"
- Bank Defendant Goldman Sachs does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Goldman Sachs's accused instrumentalities meets the claim element "the lossless encoders are selected;"
- Bank Defendant Goldman Sachs does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Goldman Sachs's accused instrumentalities meets the claim element "descriptors indicate;"

- Bank Defendant Goldman Sachs does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for Goldman Sachs's accused instrumentalities, Goldman Sachs does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"
- Bank Defendant HSBC does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of HSBC's accused instrumentalities meets the claim element "analyses of content of the data blocks;"
- Bank Defendant HSBC does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of HSBC's accused instrumentalities meets the claim element "the lossless encoders are selected;"
- Bank Defendant HSBC does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of HSBC's accused instrumentalities meets the claim element "descriptors indicate;"
- Bank Defendant HSBC does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for HSBC's accused instrumentalities, HSBC does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"
- Bank Defendant J.P. Morgan does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of J.P. Morgan's accused instrumentalities meets the claim element "analyses of content of the data blocks;"

- Bank Defendant J.P. Morgan does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of J.P. Morgan's accused instrumentalities meets the claim element "the lossless encoders are selected;"
- Bank Defendant J.P. Morgan does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of J.P. Morgan's accused instrumentalities meets the claim element "descriptors indicate;"
- Bank Defendant J.P. Morgan does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for J.P. Morgan's accused instrumentalities, J.P. Morgan does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"
- Bank Defendant Morgan Stanley does not infringe claims 15, 20, 22 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of Morgan Stanley's accused instrumentalities meets the claim element "analyzing [or analyze] content of the [or a] data block [or field of the message] to determine a data block [or field] type," "analyses of content of [the] data blocks [or fields]," "recognizing a data field type of a data field"[1] or "recognizing data field types;"[2]

---

[1] This term was construed in the in the July 9, 2012 Order Adopting the Parties' Agreed Claim Constructions (D.I. No. 506 in Case No. 11 Civ. 6696) ("Agreed Claim Construction Order") as "recognizing a data field type by analyzing the content of the data field."

[2] While "recognizing data field types" was not specifically construed in the Agreed Claim Construction Order, claim 20 of the '568 Patent, which contains that term, was listed in the claim term chart next to "recognizing a data field type of a data field," and the Court hereby adopts as

- Bank Defendant Morgan Stanley does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of Morgan Stanley's accused instrumentalities meets the claim element "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders," or "select one or more lossless encoders;"
- Bank Defendant Morgan Stanley does not infringe claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent because none of Morgan Stanley's accused instrumentalities meets the claim element "descriptor with the encoded data which identifies," "descriptors indicate" or "descriptor indicates;"
- Bank Defendant Morgan Stanley's accused decoding instrumentalities identified in paragraph 17 of the Declaration of Daniel Hirschberg In Support of Downstream Defendants' Motion For Summary Judgment of Non-Infringement of the Asserted Encoding-Decoding Claims On The Grounds That Downstream Defendants Do Not Select Encoders (D.I. 520 in Case No. 11 Civ. 6696), dated July 16, 2012, do not infringe claims 95, 97, 108 and 112 of the '651 Patent because for those accused decoding instrumentalities, Morgan Stanley does not perform (or directly control another who performs) the claim limitation "wherein

---

the construction of "recognizing data field types" the same construction that it adopted for the substantially identical term "recognizing a data field type of a data field." (*See* Agreed Claim Construction Order at 2.)

the lossless encoders are selected based on analyses of content of the data blocks;" and

- Claims 1, 7, 8 and 13 of the '747 Patent and claims 1, 4, 6, 7 and 12 of the '651 Patent are invalid under 35 U.S.C. § 112, first and second paragraphs, for failure to comply with the definiteness and written description requirements.

Further, the Court hereby enters Final Judgment that Realtime is precluded from asserting infringement under the doctrine of equivalents through its expert or any related testimony. (August 2, 2012 Memorandum and Order (D.I. No. 542 in Case No. 11 Civ. 6696).) Realtime has also represented to the Bank Defendants that it is not pursuing indirect infringement against the Bank Defendants.

Furthermore, since the effect of this Final Judgment resolves liability as to the Bank Defendants' accused instrumentalities, the Court hereby dismisses as moot and without prejudice any remaining, unresolved counter-claims and affirmative defenses raised by any party in this Action. For the purpose of clarity, the Bank Defendants reserve all rights to urge affirmance on any ground supported by the record and advanced in the district court.

Further, it is Ordered that, to the extent not granted or denied by an existing Court Order, all currently pending motions and joinders thereto are denied without prejudice, with leave to renew if an appropriate time arises.

Finally, notwithstanding the above, this Court retains jurisdiction to hear any attorneys' fees motions brought by the Bank Defendants in this Action. The Court will set a briefing schedule for those motions in due course. Accordingly, in the interim, the parties' time to file such motions under Federal Rule of Civil Procedure 54(d)(2)(B) is hereby extended to such date as the Court will set in the future.

SO ORDERED:

Dated: New York, New York
November 9, 2012

K. B. Forrest
KATHERINE B. FORREST
UNITED STATES DISTRICT JUDGE

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO,

Plaintiff,

v.

THOMSON REUTERS CORPORATION, et al.,

Defendants.

No. 11 Civ. 6698 (KBF)
No. 11 Civ. 6700 (KBF)
No. 11 Civ. 6703 (KBF)

FINAL JUDGMENT
PURSUANT TO FED. R. CIV. P. 54(b)

Plaintiff Realtime Data, LLC d/b/a IXO ("Realtime") brought the above-captioned Actions (the "Action") against Defendants Thomson Reuters Corporation ("Thomson Reuters"), FactSet Research Systems Inc. ("FactSet"), Bloomberg L.P. ("Bloomberg") and Interactive Data Corporation ("IDC") (collectively, the "Data Provider Defendants"), alleging that the Data Provider Defendants infringe U.S. Patent Nos. 7,417,568 (the "'568 Patent"), 7,777,651 (the "'651 Patent") and 7,714,747 (the "'747 Patent").

On June 22, 2012, the Court issued an Opinion and Order (*Markman*) in this Action (D.I. No. 371 in Case No. 11 Civ. 6698).

On June 27, 2012, the Court issued an Opinion and Order (Partial Motion for Summary Judgment re: Data Decompression) (D.I. No. 376 in Case No. 11 Civ. 6698) in this Action finding that claims 1, 7, 8 and 13 of the '747 Patent, and claims 1, 4, 6, 7 and 12 of the '651 Patent are invalid under 35 U.S.C. § 112, first and second paragraphs, for failure to comply with the definiteness and written description requirements.

On September 24, 2012, the Court issued an Opinion and Order (D.I. No. 456 in Case No. 11 Civ. 6698) in this Action granting certain motions for summary judgment of non-infringement filed or joined by Defendants in this Action and the related actions (with which this Action was consolidated for pre-trial proceedings) *Realtime Data, LLC d/b/a IXO v. Morgan Stanley, et al.* (Case Nos. 11 Civ. 6696, 6701, 6704) and *Realtime Data, LLC d/b/a IXO v. Thomson Reuters, et al.* (Case Nos. 11 Civ. 6698, 6700, 6703) (the "Related Actions"), and finding that all of the Data Provider Defendants' accused instrumentalities fail to meet one or more of the elements of each asserted claim of the '651 Patent.

On November 9, 2012, the Court issued a Supplemental Order granting certain additional motions for summary judgment of non-infringement filed or joined by Defendants in this Action and the Related Actions.

The Court's Opinions and Orders resolve the claims brought by Realtime in its pleadings against each Data Provider Defendant in this Action, and there being no just reason for delay, pursuant to Federal Rule of Civil Procedure 54(b), the Court hereby enters Final Judgment of non-infringement and invalidity of the specific patent claims addressed in the subject Opinion and Orders referenced above, including as follows:

- Data Provider Defendant Bloomberg does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Bloomberg's accused instrumentalities meets the claim element "analyses of content of the data blocks;"
- Data Provider Defendant Bloomberg does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Bloomberg's accused instrumentalities meets the claim element "the lossless encoders are selected;"

- Data Provider Defendant Bloomberg does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of Bloomberg's accused instrumentalities meets the claim element "descriptors indicate;"
- Data Provider Defendant Bloomberg does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for Bloomberg's accused instrumentalities, Bloomberg does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"
- Data Provider Defendant FactSet does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of FactSet's accused instrumentalities meets the claim element "analyses of content of the data blocks;"
- Data Provider Defendant FactSet does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of FactSet's accused instrumentalities meets the claim element "the lossless encoders are selected;"
- Data Provider Defendant FactSet does not infringe claims 95, 97, 108 and 112 of the '651 Patent because none of FactSet's accused instrumentalities meets the claim element "descriptors indicate;"
- Data Provider Defendant FactSet does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for FactSet's accused instrumentalities, FactSet does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;"

- Data Provider Defendant IDC does not infringe claims 95, 97, 108 and 112 of the ’651 Patent because none of IDC’s accused instrumentalities meets the claim element “analyses of content of the data blocks;”
- Data Provider Defendant IDC does not infringe claims 95, 97, 108 and 112 of the ’651 Patent because none of IDC’s accused instrumentalities meets the claim element “the lossless encoders are selected;”
- Data Provider Defendant IDC does not infringe claims 95, 97, 108 and 112 of the ’651 Patent because none of IDC’s accused instrumentalities meets the claim element “descriptors indicate;”
- Data Provider Defendant IDC does not infringe claims 95, 97, 108 and 112 of the ’651 Patent because for IDC’s accused instrumentalities, IDC does not perform (or directly control another who performs) the claim limitation “wherein the lossless encoders are selected based on analyses of content of the data blocks;”
- Data Provider Defendant Thomson Reuters does not infringe claims 95, 97, 108 and 112 of the ’651 Patent because none of Thomson Reuters’s accused instrumentalities meets the claim element “analyses of content of the data blocks;”
- Data Provider Defendant Thomson Reuters does not infringe claims 95, 97, 108 and 112 of the ’651 Patent because none of Thomson Reuters’s accused instrumentalities meets the claim element “the lossless encoders are selected;”
- Data Provider Defendant Thomson Reuters does not infringe claims 95, 97, 108 and 112 of the ’651 Patent because none of Thomson Reuters’s accused instrumentalities meets the claim element “descriptors indicate;”

- Data Provider Defendant Thomson Reuters does not infringe claims 95, 97, 108 and 112 of the '651 Patent because for Thomson Reuters's accused instrumentalities, Thomson Reuters does not perform (or directly control another who performs) the claim limitation "wherein the lossless encoders are selected based on analyses of content of the data blocks;" and
- Claims 1, 7, 8 and 13 of the '747 Patent and claims 1, 4, 6, 7 and 12 of the '651 Patent are invalid under 35 U.S.C. § 112, first and second paragraphs, for failure to comply with the definiteness and written description requirements.

Further, the Court hereby enters Final Judgment that Realtime is precluded from asserting infringement under the doctrine of equivalents through its expert or any related testimony. (August 2, 2012 Memorandum and Order (D.I. No. 419 in Case No. 11 Civ. 6698).) Realtime has also represented to the Data Provider Defendants that it is not pursuing indirect infringement against the Data Provider Defendants.

Furthermore, since the effect of this Final Judgment resolves liability as to the Data Provider Defendants' accused instrumentalities, the Court hereby dismisses as moot and without prejudice any remaining, unresolved counter-claims and affirmative defenses raised by any party in this Action. For purposes of clarity, the Data Provider Defendants reserve all rights to urge affirmance on any ground supported by the record and advanced in the trial court.

Further, it is ordered that, to the extent not granted or denied by an existing Court Order, all currently pending motions and joinders thereto are denied without prejudice, with leave to renew if an appropriate time arises.

Finally, notwithstanding the above, this Court retains jurisdiction to hear any attorneys' fees motions brought by the Data Provider Defendants in this Action. The Court will set a

briefing schedule for those motions in due course. Accordingly, in the interim, the parties' time to file such motions under Federal Rule of Civil Procedure 54(d)(2)(B) is hereby extended to such date as the Court will set in the future.

SO ORDERED:

Dated: New York, New York
November 9, 2012

K B. Forrest

KATHERINE B. FORREST
UNITED STATES DISTRICT JUDGE

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO,

Plaintiff,

-v-

MORGAN STANLEY, et al.,

Defendants.

11 Civ. 6696 (KBF)
11 Civ. 6701 (KBF)
11 Civ. 6704 (KBF)

OPINION AND ORDER

REALTIME DATA, LLC d/b/a IXO,

Plaintiff,

-v-

CME GROUP INC., et al.

Defendants.

11 Civ. 6697 (KBF)
11 Civ. 6699 (KBF)
11 Civ. 6702 (KBF)

OPINION AND ORDER

REALTIME DATA, LLC d/b/a IXO,

Plaintiff,

-v-

THOMSON REUTERS, et al.

Defendants.

11 Civ. 6698 (KBF)
11 Civ. 6700 (KBF)
11 Civ. 6703 (KBF)

OPINION AND ORDER

(MARKMAN)

KATHERINE B. FORREST, District Judge:

Plaintiff Realtime Data, LLC ("Realtime") brought nine lawsuits asserting patent infringement against approximately 18 separate defendants. The defendants break down into three general groups: (i) major financial institutions including Morgan Stanley, Bank of America Corporation, Credit Suisse Holdings (USA), Inc., The Goldman Sachs Group, HSBC Bank USA,

N.A., JP Morgan Chase & Co., etc.; (ii) stock exchanges and other trading platforms such as the entities that form what is colloquially known as the New York Stock Exchange ("NYSE") (e.g., NYSE Euronext, NYSE Arca), entities that make up what is colloquially known as the NASDAQ (e.g., Nasdaq OMX PHLX, Inc.), the Chicago Mercantile Exchange, New York Mercantile Exchange, Inc., International Securities Exchange, etc.; and (iii) companies who the Court refers to as "market information" companies and who are in the business of providing financial information to, inter alia, companies in the first two categories of defendants, such as Thomson Reuters, Bloomberg, L.P., Interactive Data Corporation, etc. The Court refers to the defendants across all nine actions collectively as "defendants." Defendants together have moved for the construction of certain terms used in the claims they are alleged to have infringed.

Plaintiff has asserted that defendants are, individually or collectively, infringing on 49 claims across three patents: U.S. Patent Nos. 7,714,747 ("the '747 Patent"), 7,777,651 ("the '651 Patent"), and 7,417,568 ("the '568 Patent" and with the '747 Patent and '651 Patent, the "patents-in-suit"). All three of the patents-in-suit relate generally to inventions that claim accelerated and reduced latency transmissions of, inter alia,

financial data using certain data compression and decompression techniques.

On June 2, 2010, these nine lawsuits were consolidated pursuant to Rule 42(a) of the Federal Rules of Civil Procedure. (Dkt. No. 186.) Discovery closed on June 4, 2012, and the matter is scheduled for trial on November 26, 2012. (Dkt. No. 396.)

Before the Court are the parties' requests for claim construction. Defendants and plaintiff have requested that this Court construe 11 terms used in the patents:[1]

1. data stream, stream of data;
2. encoder, encode, encoded, encoding;
3. decoder, decode, decoded, decoding;
4. data field type[s], data block type[s];
5. lossless, lossless encoder(s)/decoder(s);
6. selecting an encoder, the lossless encoders are selected, selecting one or more lossless encoders, select one or more lossless encoders;
7. content independent data compression;
8. content dependent data compression;

[1] Two of the eleven terms/phrases--"content independent/dependent data decompression"--is also the subject of a separate motion for partial summary judgment for failure to comply with the definiteness and written description requirements of 35 U.S.C. § 112. The Court's decision on that motion is the subject of a separate opinion.

9. descriptor with the encoded data which identifies, descriptors indicate, descriptor indicates;

10. content dependent data decompression; and

11. content independent data decompression.

The Court construes the above terms as set forth below.

DISCUSSION

Certain patents present straightforward, rather uncomplicated construction issues. For instance, this Court was once asked (for a brief time until the Court prevailed on the parties to withdraw their disagreement) to construe what the word "up" meant. However, there are a vast number of technologically complex patents that require a court to construe terms about which proposed experts who have spent decades in the field of, for example, electrical engineering, disagree.

Lingering underneath the proffered definitions of the claims to be construed lay parties' positions with respect to infringement or non-infringement: the addition or elimination of a descriptive word or phrase can make the difference between whether an action taken by a defendant runs afoul of a claim or not. Similarly, the presence or absence of a defining word can make the difference between a claim's validity: whether the claim avoids--or runs headlong--into prior art.

In the context of claim construction, a court must ignore the effect that a particular construction may have on the

outcome of the litigation and, instead, seek the correct construction. This Court has and does proceed with claim construction without regard to the impact constructions will or may have on the merits of the underlying claims in the nine consolidated lawsuits. To do otherwise would be to advertently or inadvertently usurp the role of the fact finder (which in this case is a jury) on the ultimate merits.

I. LEGAL STANDARDS

A. Claims Construction

Claim construction is a question of law for the court. Markman v. Westview Instruments, Inc., 52 F.3d 967, 979 (Fed. Cir. 1995), aff'd, 517 U.S. 370 (1996). Determining the meaning of terms within a claim assists a fact finder in making subsequent and ultimate decisions as to whether an invention has in fact been infringed, or is in fact valid. In construing the meaning of a term, the issue is not what that term would mean to an average lay person, but what that term or phrase would have meant to one "of ordinary skill in the art in question at the time the invention was made." Phillips v. AWH Corp., 415 F.3d 1303, 1313 (Fed. Cir. 2005). A court's job is, then, to try and place itself in the position of one of ordinary skill in the art of the invention(s) at issue.

There is a substantial body of law setting forth the appropriate tools with which a court should work in construing

claims, the order in which those tools should be utilized, and the weight that should be given to additional resources brought to bear on proffered constructions. A court may use intrinsic--and, if necessary, extrinsic--evidence. See Nazomi Commc'ns, Inc. v. Arm Holdings, PLC, 403 F.3d 1364, 1368 (Fed. Cir. 2005) (instructing that courts should look first to intrinsic evidence).

Intrinsic evidence includes the claims and specifications in the patent itself, as well as the patent's file history (or wrapper). The single most important source for the meaning of a term is the language of the claim itself--the language of a claim defines the scope of the patent holder's exclusive rights. Phillips, 415 F.3d at 1312. In patents with multiple claims using similar terms, such as the patents-in-suit, it is well accepted that terms in a claim should be construed consistently across claims. Southwall Techs., Inc. v. Cardinal IG Co., 54 F.3d 1570, 1579 (Fed. Cir. 1995) ("[C]laim terms must be interpreted consistently.").

One skilled in the art is "deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." Phillips, 415 F.3d at 1313. The specification is the "single best guide to the meaning of a disputed term." Id. at 1315; see also On Demand Machine Corp.

v. Ingram Indus., Inc., 442 F.3d 1331, 1338, 1340 (Fed. Cir. 2006) ("[T]he scope and outer boundary of claims is set by the patentee's description of his invention" and "the claims cannot be of broader scope than the invention that is set forth in the specification."). However, although specifications contain one or more examples of the embodiment of an invention, they need not contain every possible embodiment; therefore, courts should not read into the claims limitations based on the embodiments in the specification. See Phillips, 415 F.3d 1323; Innogenetics, N.V., v. Abbott Labs., 512 F.3d 1363, 1370 (Fed. Cir. 2008) ("[The defendant] argues that a patent can never be literally infringed by embodiments that did not exist at the time of filing. Our case law allows for after-arising technology to be captured within the literal scope of valid claims that are drafted broadly enough.").

Although terms are generally construed as they would be understood by one of ordinary skill in the art, it is possible for a patentee to have set forth a particular and different meaning for a term within a claim; in such a case, the lexicography of the patentee governs. See, e.g., Silicon Graphics, Inc. v. ATI Techs., Inc., 607 F.3d 784, 789 (Fed. Cir. 2010); Southwall Techs., 54 F.3d at 1578 ("The terms in a claim, however, are not given their ordinary meaning to one of skill in

the art when it appears from the patent and file history that the terms were used differently by the applicant.").

Several aspects of the patent prosecution history can be of significant use to a court. First, statements the patentee may have made in connection with patent prosecution can be binding. The prosecution history provides evidence of how the patentee understood and explained his invention to the patent office. Phillips, 415 F.3d at 1317; see also Kripplez v. Ford Motor Co., 667 F.3d 1261, 1266-67 (Fed. Cir. 2012) (statements made during patent prosecution proceedings are binding on the patentee and can be considered as such during claim construction); Teleflex, Inc. v. Ficosa N. Am. Corp., 299 F.3d 1313, 1326 (Fed. Cir. 2002)("[T]he prosecution history (or the file wrapper) limits the interpretation of claims so as to exclude any interpretation that was disclaimed or disavowed during prosecution in order to obtain claim allowance.").

In connection with patents that are part of an extended family of patents, a patentee's disclaimers made during prosecution are "relevant" both as a statement made with regard to the patent at issue, but also with regard to related or "sibling" patents. See Microsoft Corp., v. Multi-Tech Sys., Inc., 357 F.3d 1340, 1349-50 (Fed. Cir. 2004). Statements made by a patentee during a prosecution history prevent claim terms from becoming ever-changing as the need and situation changes.

See Southwall Techs., 54 F.3d at 1578 ("A patentee may not proffer an interpretation for purposes of litigation that would alter the indisputable public record consisting of the claims, the specification and the prosecution history, and treat the claims as a 'nose of wax.'").

In addition, the patent examiner is considered to be one of ordinary skill in the art. See St. Clair Intellectual Prop. Consultants, Inc. v. Canon Inc., 412 Fed. Appx. 270, 276 (Fed. Cir. 2011); In re Lee, 277 F.3d 1338, 1345 (Fed. Cir. 2002). Statements made by the examiner relating to how he or she understands a certain term are thus intrinsic evidence to which the court may refer when construing terms.

Extrinsic evidence includes dictionaries used by one of ordinary skill in the art, treatises and expert testimony in the form of affidavits or presented live at a Markman hearing. Phillips, 415 F.3d at 1314.[2]

B. Expert Witnesses

With respect to expert witnesses, courts apply Rule 702 of the Federal Rules of Evidence and evaluate whether the testimony is helpful and the credibility of the witness, and considers whether proffered opinions meet the standards set forth in

[2] The Federal Circuit directed that courts should use dictionaries when necessary and useful, but with appropriate caution. See Innogenetics, 512 F.3d at 1371 (citing Phillips for its cautionary language regarding elevation of dictionaries to such prominence that it focuses the inquiry on the abstract meaning of the words rather than the meaning of claim limitations).

Daubert v. Merrell Dow Pharmaceuticals, Inc., 509 U.S. 579 (1993). Rule 702 provides:

> A witness who is qualified as an expert by knowledge, skill, experience, training, or education may testify in the form of an opinion or otherwise if: (a) the expert's scientific, technical, or other specialized knowledge will held the trier of fact to understand the evidence or to determine a fact in issue; (b) the testimony is based on sufficient facts or data; (c) the testimony is the product of reliable principles and methods; and (d) the expert has reliably applied the principles and methods to the facts of the case.

Fed. R. Evid. 702. In addition, a court should not consider mere ipse dixit of an expert--that is, conclusory statements without analytical basis. Country Rd. Music, Inc v. MP3.com, Inc., 279 F. Supp. 2d 325, 330 (S.D.N.Y. 2007).

The Court will review the qualifications and testimony of both experts further below. For the moment, suffice it to say, in this matter, the Court found that both Dr. Michael Ian Shamos (for plaintiff) and Dr. James Storer (for defendants) provided helpful testimony. However, given the nature of the patents-in-suit, which relate to data compression and decompression techniques, this Court found Dr. Storer's expertise--of several decades spent focusing on compression and decompression technologies--is particularly relevant and therefore particularly useful to construing the terms at issue.

II. EVIDENCE RELEVANT TO THE CONSTRUCTIONS AT ISSUE

In connection with construing the claims at issue in this matter, the Court has referred to both intrinsic and extrinsic

evidence. The Court's primary tool was, as the law requires, the language in the claims and the specifications of the patents-in-suit, the prosecution history relating to those patents (including statements by the patent examiner), and the specifications and prosecution history of related patents.

Prior to the Markman hearing, the parties submitted technical tutorials to explain to the Court how one of ordinary skill in the art would understand the inventions set forth in the patents-in-suit. (See 11 Civ. 6696, Minute Order (Apr. 24, 2012) (regarding submission of 2 DVDs by defendants and 1 thumb drive by plaintiff).) At a conference prior to the Markman hearing, the Court notified the parties that if they wanted the Court to rely upon such tutorials as a form of extrinsic evidence (the equivalent of testimony or affidavit from an expert), they should take that into consideration in how they presented the tutorials (e.g., through an expert or through a lawyer). (Tr. 61:2-15, 62:14-20 (Apr. 16, 2012) (11 Civ. 6696, Dkt. No. 429).) If a tutorial was submitted as a type of expert submission, the other side would be (and was) provided an opportunity to cross-examine the expert at the Markman hearing itself. (Tr. 62:3-5, 62:16-20 (Apr. 16, 2012)).

Both parties put forth experts in support of their proposed constructions. Plaintiff's expert, Michael Ian Shamos, submitted declaration. (11 Civ. 6697, Dkt. No. 565 ("Shamos

Decl.").) Dr. Shamos is the Distinguished Career Professor in the School of Computer Science at Carnegie Mellon University, where he teaches graduate courses and where he was a founder and Co-Director of the Institute for eCommerce. (Shamos Decl. ¶ 2.) He also founded two computer software development companies between 1979 and 1987. (Id. ¶ 3.) He has also previously testified in cases concerning "electronic auctions and electronic payment system[s]." (Id. ¶ 6.)

For defendants, Dr. James Storer provided, as mentioned above, a video tutorial as well as live testimony. Dr. Storer is currently a professor of computer science at Brandeis University, where he teaches graduate and undergraduate level courses and where the focus of his research has been data compression. Tr. 71:11-20 (May 4, 2012). Dr. Storer published his first book on data compression, Data Compression Methods & Theory, in 1988. Tr. 72:4-9. Dr. Storer was also intimately involved in the creation of the IEEE Data Compression Conference which first took place in 1991, and has been its chair since its inception--and where he is a review of all papers presented at the conference. Tr. 73:11-15, 73:21-74:6. It was clear from his testimony that Dr. Storer has been researching--and involved in the development of the field of--data compression since it "exploded" in the late 1980s. Tr. 73:2-10.

Both experts appeared live at the Markman hearing, answered questions posed by the Court, and were subject to cross-examination.

Defendants chose to designate the portion of their video tutorial narrated by Dr. Storer as proffered expert testimony. Finding Dr. Storer qualified as an expert, one skilled in the art, and his testimony helpful, this Court has, where indicated, relied upon that video tutorial, among other forms of evidence, when it needed to refer to extrinsic evidence to construe a term.

Plaintiff chose to submit their tutorial from Dr. Shamos only as background and not as evidence. As an evidentiary matter, therefore, this Court has not considered plaintiff's tutorial to constitute anything more than legal argument.

III. THE CONSTRUCTIONS

A. Data Stream/Stream of Data

The parties agree on the first portion of the proposed construction for the phrases "data stream" or "stream of data": "one or more data blocks transmitted in sequence." Defendants, however, have suggested additional language which would follow this initial statement: "from an external source whose characteristics are not controlled by the data encoder or decoder." (11 Civ. 6696, Dkt. No. 489.) The Court agrees with defendants' proposed construction. The Court's determination is

based upon the fact that Realtime has previously asserted either the substance of, or the precise words of, defendants' construction in multiple re-examination proceedings relating to one of the patents-in-suit (i.e., the '568 Patent) as well as other related patents.

The specification of the patents that were subject to the reexamination have a specification that is identical to the '747 patent-in-suit (those patents are the '506, '761, and '992 Patents). In turn, the specification of the '747 Patent is incorporated by reference into the two other patents-in-suit-- the '568 and the '651 Patents.

Exhibit 10 to the Declaration of Michael Murray in Support of Defendants' Opening Brief on Claim Construction Issues ("Murray Decl." (11 Civ. 6697, Dkt. No. 563)) comprises a series of submissions made by plaintiff and various experts retained by plaintiff setting forth definitions of "data stream" consistent with that proffered by defendants here. For instance, Dr. George T. Ligler submitted five separate declarations spanning the period from August 2009 to December 2010, and Dr. James Modestino submitted four declarations spanning the period from March 2010 to July 2011. (Murray Decl. ¶ 11 & Ex. 10.)

Plaintiff seeks to distinguish those declarations on the basis that they are defining the phrase "receiving a data stream." That argument reads the declarations too narrowly and

ignores the plain language in the declarations themselves. It is certainly true that the declarations refer to the term "receiving"--but they do so in the context of defining the term "data stream". An issue on reexamination was, in part, that certain prior art used the term "data stream" in manner that might have otherwise captured plaintiff's invention. In describing why and how its invention was distinguishable from prior art, plaintiff explained that the concept of "receiving" was imbedded within the use of the term "data stream."

The Modestino Declaration submitted on March 15, 2010 (attached as part of Exhibit 10 to the Murray Declaration) is illustrative. (See Murray Decl. Ex. 10.) In his declaration Dr. Modestino stated, "The context provided by the '506 specification [as stated above, the specification of the '747 patent-in-suit is the same as the specification of the '506 patent, and the '568 and '651 patents-in-suit are incorporated by reference into the '747 Patent] makes it clear to one of ordinary skill in the art that the use of the term 'data stream' was intended by the inventor to convey a continuous stream of data elements received from or transmitted." (Murray Decl. Ex. 10 (Modestino Decl. (Mar. 15, 2010) at 3).) Dr. Modestino then discusses how compression occurs once data is "received" and states that it was the "inventor's clear intent to convey that by the term 'data stream' a continuous stream of data elements

to be received or transmitted was intended." (Id.) He then offers the following definition:

> A person of ordinary skill in the art would consider the phrase 'receiving a data stream' to imply a stream of data transmitted from a source (whose characteristics are therefore not controlled by the data compression system) and received at the input of a system or device. As noted above, in the context of the '506 patent, one of ordinary skill in the art would consider a data stream as a continuous stream of data elements to be received or transmitted. Furthermore, the process of receiving a data stream, represented by a sequence of data blocks, is generally considered a passive one requiring no control over the characteristics of the received data stream by the receiver . . . .

(Id.) Dr. Modestino then stated, "if the disclosed compression system did participate in the transmission of the received data stream it would be an important feature of the invention and would require some discussion in order for someone to successfully practice the invention." (Id.)

Similarly, in Realtime's "Reply to the [PTO] Action in Reexamination," dated July 8, 2009, it draws distinctions between its invention and the prior art of Baker, asserting that plaintiff's invention there at issue, a "compression card," uses the term "data stream" to incorporate the process of passive receipt. (Murray Decl. Ex. 10 (Reply to Office Action in Reexamination (July 8, 2009) at 70).) Other patents that were part of reexamination proceedings incorporate the disclosure of the great-grandparent of the '747 Patent--declarations submitted in those reexaminations propose similar constructions of "data

stream." Exhibit 10 of the Murray Declaration contains a number of similar examples.

Realtime has argued that it should not be held to prior positions set forth in these declarations because it "retracted" the legal arguments in a filing made with the Patent and Trademark Office ("PTO"). According to Realtime, by retracting its arguments, it made it clear (publicly) it would not be relying on those positions. Realtime concedes that it was procedurally unable to remove the expert declarations (contained at Exhibit 10 of the Murray Declaration) filed on the PTO's public docket. Tr. 205:17-25 (May 4, 2012) (11 Civ. 6697, Dkt. No. 616) (Markman Hr'g). Realtime's retraction was therefore only partial: it retracted legal arguments and not factual positions made by those skilled in the art.

It is certainly reasonable for this Court and the public to rely upon the declarations of those who profess to be skilled in the art regarding their opinions as to the meaning of certain terms, particularly with respect to "sibling patents"--i.e., patents deriving from the same parent patent. Presumably, neither skill in the art nor opinions based thereon change because of a party's retraction or legal arguments. Accordingly, the Court finds that plaintiff's assertions in declarations of related patents are alone sufficient to support

adoption of defendants' construction. There is, however, additional support for such adoption.

Plaintiff argues that its construction is consistent with that which it put forth and which the court adopted in litigation involving related patents, Realtime v. Packeteer, No. 6:08-cv-144-LED-JDL (E.D. Tex.) (the "Packeteer litigation"). It is demonstrably the case that plaintiff did propose the same construction in Packeteer that it proposes here. However, its proposed construction in Packeteer occurred in July 2009--before the declarations submitted in the reexamination proceedings discussed above.

Moreover, plaintiff has also commenced litigation against a number of companies in which it is asserting related patents, Realtime Data, LLC v. MetroPCS Texas, LLC, No. 6:10-cv-49 (E.D. Tex.) (the "MetroPCS litigation"). There, plaintiff has proposed a construction of "data stream" consistent with that set forth in its expert declarations referred to above. In a similar effort to distinguish its position in MetroPCS from that here, plaintiff argues that the term in MetroPCS is "receiving a data stream" and not merely "data stream." Tr. 197:16-198:15 (May 4, 2012).

Positions taken as to claim construction in MetroPCS are certainly relevant to claim construction here. That litigation involves the '506 Patent--which, as stated above, shares a

specification with the '747 patent-in-suit, and the '568 and '651 patents-in-suit are incorporated by reference into the '747 Patent. At the April 12, 2012 Markman hearing in the MetroPCS litigation, Realtime's counsel there (different from counsel of record in the matter before this Court) argued that "the claim system cannot actively control the characteristics of the received data if that data is still part of a data stream," see Tr. 207:5-7; (see also Murray Decl. Ex. 40 at 66:5-13); and also that "the data must be transmitted from an external source and cannot be retrieved from an internal part of the receiving system," see Tr. 207:7-9.

As a matter of law, claim terms appearing in related patents should generally be construed consistently across them. See Omega Eng'g Inc. v. Raytek Corp., 334 F.3d 1314, 1334 (Fed. Cir. 2003).

B. Encoder(s), Encode, Encoded, Encoding

Plaintiff has offered the following construction for the terms encoder(s), encode, encoded, and encoding: "hardware or software that performs data compression." (Dkt. No. 489.)

For the term "encoder," defendants have slightly different language regarding compression and add a significantly different concept thereafter: "hardware or software that compresses data by converting the contents of a data block (or data field) into a coded representation of those contents." For the terms

"encode/ed/ing," defendants' construction is "compress/ed/ing by converting the contents of a data block (or data field) in a coded representation of those contents." (Dkt. No. 489.)

At its core, the difference between the two constructions is that plaintiff's construction makes the term "encoding" (and derivations thereof) synonymous with "compression." That is, all compression is encoding (but it is not at all clear that even plaintiff would argue that all encoding is the equivalent of compression). Defendants' position requires that there be a "coded" representation of the contents of what is being compressed. Put another way, for the defendants, compression may encode, but it may not--and to encode there must be some form of coding. The Court agrees with defendants' construction.

First, during the patent reexamination process for a related patent (the '274 Patent), Examiner Hughes found that "to encode" requires some form of coding. (See Murray Decl. Ex. 17 at 33-34.) Importantly, the '274 Patent has the same specification as the '568 and '651 patents-in-Suit. According to Examiner Hughes, no change in the representation of the data is "not encoding." (Id.) As a matter of law, Examiner Hughes acts from the viewpoint of one skilled in the art. See St. Clair Intellectual Prop. Consultants, Inc., 412 Fed. Appx. at 276; In re Lee, 277 F.3d at 1345. Her statements, as part of the prosecution history of related patents, are intrinsic

evidence upon which the Court can rely. See Phillips, 415 F.3d at 1317.

Plaintiff argues that this Court should ignore the statements by Examiner Hughes because it had withdrawn the claims to which these remarks referred, but that Examiner Hughes had ignored such withdrawal and proceeded to disagree with Realtime's arguments. This Court is not persuaded by that argument. Statements of a patent examiner in reexamination are necessarily part of the prosecution history of a patent; that is a separate point from whether a party has put forward a position in the context of prosecuting a patent that is later adopted. Furthermore, in connection with resisting a request for a stay of this matter, plaintiff supported Examiner Hughes' qualifications as one skilled in the art. (See Decl. of Robert Molitors in Opp'n to CME's Mot. for a Stay (11 Civ. 6697 (under seal)) ¶¶ 18-21.)

In addition, however, the extrinsic evidence is also supportive of defendants' position. First, the specialized dictionaries used by those skilled in the art also use the word "code". (See Murray Decl. Exs. 19-20.) In addition, Realtime itself defined this term (and derivations thereof) as requiring coding in its own definitions of those terms in litigation papers in this case. (See Murray Decl. Ex. 35 ("encode . . . means to express data or a message in terms of a code . . .").)

Finally, this Court credits the testimony of Dr. Storer who has spent most of his career in the field of compression/decompression techniques. Dr. Storer testified credibly that one skilled in the art would understand "encode," "encoding" and derivations thereof, as requiring some form of "coding," not simply compression.

The most significant example to which both plaintiff's and defendants' expert returned was whether the act of "throwing data away" as part of a compression process could constitute a form of "encoding". See, e.g., Tr. 92:10-13 (May 4, 2012) (Storer: "But the act of throwing something away, just that act by itself, the act of throwing away data, that's not encoding. The encoding is the fact that you actually produced something that you could decode and get back the original."). According to Dr. Storer, the word "encoding" "has had a very well-understood technical meaning since 1948, when Shannon at Bell Labs started talking about coding theory. It certainly means taking an input and producing an output, an output that typically is different." Tr. 96:16-19 (May 4, 2012).

Plaintiff's witness, Dr. Shamos, agreed that "[o]utside of these patents, encoding does not necessarily mean compression at all." Tr. 25:2-3 (May 4, 2012). He also agreed that when data is encoded, its representation has been changed. Id. The difference between Dr. Shamos and Dr. Storer is that for Dr.

Shamos, simply making a data block smaller (e.g., compressing it) equates in the patents-in-suit with a type of changed representation that is "encoding." This Court construes "coding" as requiring a change in representation, not simply "no change." This means that simply making something smaller, or throwing data away, is not coding. That is consistent with Examiner Hughes and Dr. Storer; it is frankly unclear whether it is consistent with Dr. Shamos who seemed to take various positions on this issue. To the extent that Dr. Shamos testified that throwing data away or "no change" could constitute "coding" or "encoding," the Court declines to credit that testimony as reliable.

The parties have suggested constructions for the terms "decoder(s)," "decode," and derivations thereof that are the mirror opposites of their "encode" equivalents. Therefore, this Court construes decoder(s), decode and derivations thereof consistent with defendants' position: "Decoder: hardware or software that decompresses data by reconstructing encoded data"; and "Decode/ed/ing: decompress/ed/ing by reconstructing encoded data."

C. Data Field Type[s], Data Block Type[s]

Plaintiff urges that the term "data field type" or "data block type" be construed to mean "an attribute or characteristic of the data block or data field." (Dkt. No. 489.) Defendants,

on the other hand, urge a very different construction: "categorization of the data in the field (or block) as one of ASCII, image data, multimedia data, signed and unsigned integers, pointers, or other data type." (Id.)

The essential difference between the two constructions is that plaintiff's is broad enough to capture any characteristic or any attribute of data. In other words, for plaintiff, a single parameter or value of data could fall within the definition of "data field type" or "data block type." Plaintiff urges that defendants' construction requires a content check or analysis.

According to defendants, their proposed construction is based upon--and limited to--the type of data within a data block--that is, its content. Therefore, for defendants, a "data block type" is synonymous with a "type of data block"; a "data field type" is synonymous with a "type of data field." Defendants' proposed construction refers to content as types such as ASCII, image data, "or other data type." According to defendants, their inclusion of the phrase "or other data type" allows for an expansive reading of this term to include data types that might not be listed but would be fairly encompassed by the invention. The Court construes the term consistent with defendants' construction.

Throughout the patent specifications for the patents-in-suit, it is clear that the type of data is what is being referred to when the phrase "data type" is used. There are too many examples of this to list them all. The only place that the words "data **block** type" or "data **field** type" are used is in the claims. As a matter of law, it is proper for the Court to look to the specification to construe the claims; the claims themselves do not shed any independent light on the issue. Review of the specifications leaves no doubt that defendants' proposed construction is the proper one. This was confirmed by Dr. Storer, as one skilled in the art.

For instance, the term "data field type" is used in claim 1 of the '568 Patent. There, the term is used to describe the compression technique as "recognizing a data field type of a data field in the data stream, wherein the data field is included in a packet." Claim 20 states, in part, "outputting an executable file that is used to process a stream of data by recognizing data field types in the data stream and applying encoders associated with the recognized data field types to encode the data stream." Both claims clearly reference content analysis of the type suggested by defendants.

In claim 14 of the '747 Patent, the term is used in the following manner: "analyzing content of the data block to determine a data block type"--the claim then proceeds to

describe the selection of the encoder based on that analysis (claim 19 of the '747 Patent describes "data block type" similarly). In claim 1 of the '651 Patent the term is used to mean, among other things, "wherein the description file comprises data field types and associated lossless decoders." In both cases, again, the analysis of the content and the use of that information as to type of data is what is being claimed.

Plaintiff argues that a "data block type" or "data field type" can constitute simply an attribute of data--that is, the expression of even a value of data. Dr. Storer testified credibly as one skilled in the art that checking a value is not determining a data block type or data field type. The intrinsic evidence does not support the narrow construction proposed by plaintiff. It is clear from the intrinsic evidence alone, and further supported by the credible extrinsic evidence, that "data field type" or "data block type" refers to a content categorization which is then identical to the invention.

D. Lossless, Lossless Encoder/decoder[3]

Both parties concede that the compression/decompression of financial data must be lossless. The dispute over construction of the term "lossless" centers upon whether the phrasing plaintiff has proposed can, should or does leave some ray of light for encoding/decoding to be equivalent to lossless techniques.

The constructions offered by the parties call into question whether a "lossless" encoder or decoder is one that requires that the encoder enable the decoder to present data that is identical to that which was encoded, versus an "exact representation" of what was encoded. The distinction is whether a process which encodes data by dropping certain data, but which indicates what has been dropped (say, ten zeros), and therefore is an "exact representation" of the original data (if for instance, the ten zeros are not meaningful to the recipient), is "lossless"--or whether "lossless" requires identicality in the most objective sense.

---

[3] Plaintiff has moved for leave to supplement the record with regard to the term "lossless." (Dkt. No. 469.) Plaintiff argues that until the Markman, it had not understood that any argument was being made that the encoding/decoding techniques set forth in the inventions were "lossy." The Court finds this argument unpersuasive and denies the motion. The parties certainly knew that the terms "lossless" was one of those to be construed--and that the essential difference between the parties reduced to whether identicality was required in the most traditional sense. Indeed, Dr. Storer's video tutorial itself laid out the distinction which defendants' argued at the Markman hearing. Resolution of that issue determines the appropriate construction--not whether in fact the merits would one day show, or not, that the techniques are "lossy".

There is nothing about the claims or specifications which indicate that the patentee was defining lossless in a manner unique to his patent(s). Indeed, the patentee repeatedly refers to lossless compression techniques "well known in the art." The claims and specifications indicate that the term "lossless" is used as one of ordinary skill in the art would use the term. Accordingly, this Court looks to the intrinsic and extrinsic evidence to determine what the term "lossless" means to one of ordinary skill in the art at the time the invention was made.

The Court finds that the extrinsic evidence indicates that the term "lossless" has the specific meaning of "identical" to one skilled in the art at the appropriate time. According to one skilled in the art, Dr. Storer, and a number of references to which defendants refer, that term means that when an encoder is lossless, literally and truly, <u>nothing</u> at all is lost; put another way, a "lossless" process exercises zero discretion as to whether the data is required or even useful, it must be identical. An encoder cannot, therefore, choose what data to drop and to represent it otherwise, and still be "lossless"--that changes the typical and accepted meaning that one of ordinary skill in the art would attribute to the term.[4] It certainly might be the case that a type of "lossless equivalent"

[4] Anything short of this would or could require subjective judgments regarding whether particular content is meaningful. Lossless techniques are not intended to have subjectivity in representation. (<u>See, e.g.</u>, Storer Tutorial at 21.)

encoding technique could be invented--but the term "equivalent" would then specify that the lossless-ness would not in fact be literal. That term was not used here--the term as understood by those skilled in the art "lossless" was--and that term has but one meaning: "identical."

In the specification of the '747 Patent (2:6-9), "lossless" is defined: "lossless data compression techniques provide an exact representation of the original uncompressed data. Simply stated, the decoded (or reconstructed) data is identical to the original unencoded/uncompressed data." (Emphasis added.) That language indicates that "exact representation" does not allow for anything less than identical. Thus, to the extent that plaintiff has proffered a construction that suggests just that, it goes beyond the specification itself.

In addition, the extrinsic evidence supports identicality and not "mere representation" (when allowing for anything less than identicality). In three texts on data compression, lossless is described with different words but to the same effect:

1. In Compression in Video and Audio, the author states, "In lossless encoding, the data from the expander are identical bit-for-bit with the original source data . . . ." (Murray Decl. Ex. 23).

2. In The Data Compression Book, the author states, "Lossless compression consists of those techniques

guaranteed to generate an exact duplicate of the input data stream after a compress/expand cycle." (Murray Decl. Ex. 24).

3. In Video Compression: Fundamental Compression Techniques and an Overview of the JPEG and MPEG Compression Systems, the author states, "[in lossless decompression] we get the same data we started with, exactly and precisely." (Murray Decl. Ex. 25.)

As Dr. Storer stated during the tutorial submitted to this Court,

> Lossless and Lossy are terms that are well known terms in the data compression area and have very particular and specific meanings. Lossless compression is compression that preserves each and every bit of the original data. If the results of decompression have even a single bit changed from the original, even if that bit has no importance, no matter how close the new representation is to the original, it is simply not lossless. Lossless is an absolute term that has been understood as such by those in the art, without ambiguity, for decades. Lossy compression, on the other hand, only preserved an approximation of the original.

(Storer Tutorial at 21.)

E. Selecting an Encoder, the Lossless Encoders are Selected, Selecting One or More Lossless Encoders, Select one or More Lossless Encoders

"In any compression or decompression system, it is critical that the decoder know how the data was encoded so that the decoder can use the right decompression algorithm on the encoded data. Otherwise, the result will be corrupt." (Storer Tutorial

at 28.) The parties offer disparate constructions, the essential difference between them being the temporal moment of encoder selection. Plaintiff supports variations of a construction such as "choosing hardware or software that performs data compression," "techniques, hardware or software that provide exact representation of the original uncompressed data are chosen", and "choosing one or more techniques, hardware or software that provide an exact representation of the original uncompressed data." (Dkt. No. 489.) Plaintiff's construction therefore leaves open the question of when the encoder is selected.

Defendants proffer: "choosing (or choose) an encoder (or lossless encoders) during the compression process based on analyses of the content of the data blocks (or data fields)." (Id.)

The intrinsic and extrinsic evidence supports defendants' construction, which this Court adopts. The Court finds that it will be important to the jury to have this additional clarification when determining fact issues at trial. First, the claims of both the '747 and '651 Patents require that the encoder selection be based on "analyses of content of the data block/fields." ('747 Patent at 26:28-30, 27:2-4, 27:50-53, 28:29-32; '651 Patent at 24:66-25:2, 25:55-58, 26:35-38, 27:39-

42, 29:9-12.) Thus, it must be that encoder selection is proximate to content analysis.

Second, during the reexamination proceedings, plaintiff also stated, "The '651 patent clearly supports that the content of the data blocks must be analyzed to select the proper encoder." (Murray Decl. Ex. 8 at 24.) In distinguishing prior art, plaintiff stated, "Carr fails to analyze a data field in order to recognize the data field type . . . . Therefore, Carr fails to disclose selecting an encoder associated with the recognized data field type." (Murray Decl. Ex.7 at 24.) Again, in both of those instances, selection is proximate to content analysis. Content analysis occurs only once the data stream has been received--it is not far removed in time. It is a necessary piece of determining whether content dependent/independent compression occurs; and therefore, which encoder is selected as most useful.

Given the language in the specifications and the prosecution history, it is appropriate and necessary to construe the selection of the encoder as occurring <u>during</u> the compression process. To suggest that the encoder selection could occur well before that, would be to eliminate all references to the words and concepts repeatedly referring to content analysis and reference throughout in the specification and prosecution history--this would not be appropriate. <u>See</u> <u>Phillips</u>, 415 F.3d

at 1315 (words in the specification must read as part of a "fully integrated written instrument"). Such a construction would also create unnecessary ambiguity regarding how content dependent/independent compression relates to the selection of the encoder.

Here, again, plaintiff argues that its positions in the prosecution history for the '568 Patent should be ignored since they were not persuasive and withdrawn. As discussed above, that argument is not compelling. As a matter of law, positions taken before the PTO need not be successful in order to be binding; and withdrawing arguments does not eliminate the fact that the positions were taken by the patentee as one skilled in his own art.

There was extensive argument at the Markman hearing regarding whether there is a philosophical question regarding when the compression process begins. This Court finds that based on the claims, specifications and testimony of Dr. Storer (which the Court credits), it is clear that when the claims use the phrase "selecting an encoder" it is necessarily done based on an analysis of the content of the data block, during the compression process.

F. Content Independent Data Compression/Content Dependent Data Compression

The constructions offered by the parties as to the terms "content independent data compression" and "content dependent

data compression" have little to do with how to understand those terms--in other words, the parties are not truly debating what the terms mean. Rather, their debate is one of chronology: whether content dependent data compression occurs first, and, therefore whether content independent data compression only follows if that first technique is unsuccessful.

Accordingly, claims construction in this instance--which will certainly assist the jury in making its factual determinations later--requires deciding "when" the dependent/independent compression occurs.

Defendants offers the following construction for "content independent data compression": "compression that is applied to input data <u>that is not compressed with</u> content dependent data compression, the compression applied using one or more encoders without regard to the encoder's (or encoders') ability to effectively encode the data block type (<u>or</u> data field type)." (Dkt. No. 489 (emphasis added).) Plaintiff argues that the term means: "compression that is applied using one or more encoders without regard to the encoder's (or encoders') ability to effectively encode the data block type (<u>or</u> data field type)." (Dkt. No. 489.)

For "content dependent data compression," plaintiff construes the term as "compression that is applied using one or more encoders selected based on the encoder's (or encoders')

ability to effectively encode the data block type (or data field type)," while defendants construe it as "compression that is applied to input data that is not compressed with content independent data compression, the compression using one or more encoders selected based on the encoder's (or encoders') ability to effectively encode the data type of the data block." (Dkt. No. 489.)

The claims and specifications, along with the credible testimony of Dr. Storer as one skilled in the art who can interpret that intrinsic evidence, makes it clear that content dependent compression occurs first--and if the content is recognized (i.e., if the data is recognized as ASCII, an appropriate encoder is then selected), content independent compression occurs if content is not recognized. The Court has tried mightily to read the claims and specifications to allow for the simultaneous or seriatim application of both compression techniques, but has found no support for such a construction. While the '651 and '568 Patents do have some usage of the word "sequential" (see, e.g., '651 Patent at 15:55; '568 Patent at 16:47), that word is used in connection with applications of a compression technique to portions of data blocks with different content type (e.g., ASCII and MPEG); and cannot reasonably be read to describe a data block first going through content

independent compression and then going through content dependent compression; and of course, this would make no sense.

Plaintiff seems to suggest, however, that the opposite might occur--that content dependent compression might be tried and then content independent compression tried--but that ignores the clear step of content analysis that precedes the compression process, which step determines which paths the data follows: into dependent or independent compression. (See, e.g., '747 Patent at claim 1; '651 Patent at claim 1; '568 Patent at claim 1.)

The phrases "content independent/dependent decompression" are, according to plaintiff, the reverse of the compression technique. According to defendants, these phrases are indefinite under 35 U.S.C. § 112. That argument is, as mentioned above, the subject of a separate summary judgment motion. The Court declines to construe these terms at this time, but addresses it in a separate decision on the summary judgment motion.

G. Descriptor with the Encoded Data which Identifies, Descriptors Indicate, Descriptor Indicates

As part of the invention at issue, neither party disputes that a descriptor is associated with a data block (for instance, to indicate which compression technology was used). The issue this Court must determine is whether that descriptor must be

physically attached or appended to the data, or can be associated with the data in some other fashion.

According to plaintiff, the proper construction is "a data token with the encoded data which identifies" or "data tokens indicate" or "data token indicates." Only the first of these constructions has the data token "with" the encoded data.

Defendants suggest the following construction: "recognizable data that is appended to the encoded data for specifying."

This Court finds that the claims and specifications make it clear that the descriptor must be "with," in the sense of "attached to," the encoded data. Thus, constructions which suggest that the descriptor can be detached from, or found elsewhere than "with" the encoded data, are not appropriate. The Court therefore adopts defendants' proposed construction.

The Court's construction is consistent with multiple references in the claims themselves in which the descriptor is indicated as being "with" or "appended to" the data. Moreover, seven flow charts of the '747 Patent (Figs. 3b, 5b, 7b, 10b, 14C, 16C and 18C), all show the step of "**append** null descriptor to an unencoded input data block" or "**append** corresponding descriptor" to an encoded data block, followed in each case with the step of "output . . . data block with . . . descriptor." While it is certainly true that plaintiff is not limited to the

embodiments set forth in the specifications, it is limited to the invention set forth in the specifications and claims. There is no support in the specifications or claims for the descriptor to be completely detached from the data block.

There was a fair amount of debate at the Markman hearing as to whether use of the word "append" could not be a proper part of the construction since it suggests that the descriptor would follow or come after the data had arrived at the decoder. According to Dr. Shamos, to one skilled in the art, "append" means at the end and "prepend" means at the beginning. Tr. 60:22-25 (May 4, 2012).

First, the specifications use the word "append" numerous times in connection with the placement of the descriptor. (See, e.g., '747 Patent at 4:13-15, 17-18; 7:62-67; in the '651 Patent: 16:36-38.) Second, Dr. Storer testified credibly that the actions described in the invention are occurring so quickly that it is irrelevant whether the descriptor is at the beginning or end of the data block, it is being processed at the same "time" either way. See Tr. 145:24-146:3, 146:14-22. In other words, it is not the placement of the descriptor which is of importance, but the fact that the descriptor must be with the encoded data in order for the invention to work in any of the manners described in or fairly encompassed by the specification.

This is fully consistent with the intrinsic evidence as well as the credible testimony of Dr. Storer.

Defendants also offer a construction of the terms "Indicates or Indentifies" as used in this phrase. The specifications use these words interchangeably and this Court construes them as synonymous. (See e.g. '747 Patent at 8:53-56, 18:63-66; '651 Patent at 16:23-26, 16:28-30.)

CONCLUSION

The claims at issue are construed as set forth above. For purposes of clarity, the Court provides the construction for each of the disputed terms (except those that are the subject of a separate summary judgment motion) below:

data stream, stream of data	One or more blocks transmitted in sequence from an external source whose characteristics are not controlled by the data encoder or decoder
encoder, encode, encoded, encoding	Encoder: hardware or software that compresses data by converting the contents of a data block (or data field) into a coded representation of those contents Encode/ed/ing: compress/ed/ing by converting the contents of a data block (or data field) into a coded representation of those contents
decoder, decode, decoded, decoding	Decoder: hardware or software that decompresses data by reconstructing encoded data Decode/ed/ing: decompress/ed/ing by reconstructing encoded data
data field type[s], data block type[s]	Categorization of the data in the field (or block) as one of ASCII, image data, multimedia data, signed and unsigned

	integers, pointers, or other data type
lossless, lossless encoder(s)/decoder(s)	Technique, software, or hardware that fully preserves the original unencoded data such that the decoded data is identical to the original unencoded data
selecting an encoder, the lossless encoders are selected, selecting one or more lossless encoders, select one or more lossless encoders	Choosing (or choose) an encoder (or lossless encoders) during the compression process based on analyses of content of the data blocks (or data fields)
content independent/dependent data compression	Content independent data compression: compression that is applied to input data that is not compressed with content dependent data compression, the compression applied using one or more encoders without regard to the encoder's (or encoders') ability to effectively encode the data block type (<u>or</u> data field type) Content dependent data compression: compression that is applied to input data that is not compressed with content independent data compression, the compression using one or more encoders selected based on the encoder's (or encoders') ability to effectively encode the data type of the data block
descriptor with the encoded data which identifies, descriptors indicate, descriptor indicates	Recognizable data that is appended to the encoded data for specifying

Plaintiff's motion for leave to supplement the record on claims construction is DENIED.

The Clerk of the Court is directed to terminate the motion at Docket No. 469 in 11 Civ. 6696, Docket No. 618 in 11 Civ. 6697, and Dkt. No. 349 in 11 Civ. 6698.

SO ORDERED:

Dated: New York, New York
June 22, 2012

K B. Forrest
KATHERINE B. FORREST
United States District Judge

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

--------------------------------------- X

REALTIME DATA, LLC d/b/a IXO,

Plaintiff,

-v-

MORGAN STANLEY, et al.,

Defendants.

11 Civ. 6696 (KBF)
11 Civ. 6701 (KBF)
11 Civ. 6704 (KBF)

--------------------------------------- X

REALTIME DATA, LLC d/b/a IXO,

Plaintiff,

-v-

CME GROUP INC., et al.

Defendants.

11 Civ. 6697 (KBF)
11 Civ. 6699 (KBF)
11 Civ. 6702 (KBF)

--------------------------------------- X

REALTIME DATA, LLC d/b/a IXO,

Plaintiff,

-v-

THOMSON REUTERS, et al.

Defendants.

11 Civ. 6698 (KBF)
11 Civ. 6700 (KBF)
11 Civ. 6703 (KBF)

--------------------------------------- X

OPINION AND ORDER
(Partial Motion for Summary Judgment re Data Decompression)

KATHERINE B. FORREST, District Judge:

Plaintiff Realtime Data, LLC d/b/a IXO ("Realtime") has filed nine separate lawsuits against more than 30 separate defendants. Realtime asserts that defendants (i.e., all defendants in the above-captioned actions) are, individually or collectively, infringing on 49 claims across three patents: U.S. Patent Nos. 7,714,747 ("the '747 Patent"), 7,777,651 ("the '651

Patent"), and 7,417,568 ("the '568 Patent" and with the '747 Patent and '651 Patent, the "patents-in-suit"). Both the '747 and the '651 Patents are continuations, in part, of prior applications. The '747 Patent traces its lineage back to an original application filed on December 11, 1998 (referred to as the '024 Patent, which issued in 2001 as the '761 Patent), and the '651 Patent traces back to a provisional application filed on October 3, 2000.

Defendants[1] have now moved for summary judgment on the grounds that the following claims are invalid for failure to meet the definiteness and written description requirements of 35 U.S.C. § 112: claims 1, 7, 8 and 13 of the '747 Patent and claims 1, 4, 6, 7 and 12 of the '651 Patent (collectively, the "claims at issue").[2]

---

[1] The original moving defendants were the "Bank Defendants"--BNY ConvergEx Group LLC, BNY ConvergEx Execution Solutions LLC, Credit Suisse Holdings (USA), Inc., Credit Suisse Securities (USA) LLC, The Goldman Sachs Group, Inc., Goldman Sachs & Co., Goldman Sachs Execution & Clearing, L.P., HSBC Bank USA, N.A., HSBC Securities (USA), Inc., J.P. Morgan Chase & Co., J.P. Morgan Securities, Inc., J.P. Morgan Clearing Corp., Morgan Stanley, and Morgan Stanley & Co., Inc. (See 11 Civ. 6696, Defs. Mem. of Law in Supp. of Mot. for Partial Summ. J. (Dkt. No. 429) at n.1.) On April 4, 2012, the "Exchange Defendants"--NYSE Euronext, NYSE ARCA, Inc., NYSE AMEX, LLC, Securities Industry Automation Corporation, Options Price Reporting Authority, LLC, International Securities Exchange, Boston Options Exchange Group LLC, CME Group Inc., Board of Trade of the City of Chicago, Inc., New York Mercantile Exchange, Inc., Chicago Board Options Exchange, Incorporated, BATS Trading, Inc., NASDAQ OMX Group, Inc., and NASDAQ OMX PHLX, Inc.--joined in this motion (11 Civ. 6697, Dkt. No. 574), as did the "Data Provider Defendants"--Thomson Reuters Corporation, Factset Research Systems, Inc., Bloomberg L.P., Interactive Data Corporation, Penson Worldwide, Inc. and Nexa Technologies, Inc. (11 Civ. 6698, Dkt. No. 317).

[2] Defendants have not moved with respect to any claims asserted in the '568 Patent. The Exchange Defendants filed a separate, additional motion for partial summary judgment (in which the Bank and Data Provider Defendants joined) which, inter alia, sought to invalidate certain claims of the '568

The claims at issue on this motion relate to decompressing compressed data using either "content dependent data decompression" or "content independent data decompression." In support of their motion, defendants have submitted a declaration from James Storer, Ph.D, an expert in data compression and decompression. ("Storer Decl." (Dkt. No. 431).) In opposition to the motion, plaintiff has submitted a declaration from Michael Ian Shamos, J.D., Ph.D., an individual with expertise in the general area of electronic commerce. ("Shamos Decl." (Dkt. No. 447).) Although dueling expert declarations frequently create questions of fact which make summary judgment inappropriate, in this instance, the nature of the question before this Court allows for summary judgment despite the presence of dueling expert declarations. To wit, the expert declarations do not create a triable issue of fact--rather, they clarify that there is none. In other words, on all the critical points the experts agree or do not disagree.

The written description provision in 35 U.S.C. § 112 requires that the applicant demonstrate the he was in possession of the invention at the time the he submitted the relevant patent application. In connection with the '747 Patent, that would have been in 1998; in connection with the '651 Patent that would have been in 2000. In addition, section 112 requires that

Patent. The Court denied that motion in a separate decision issued on June 26, 2012.

the description contain sufficient information such that those of ordinary skill in the art can understand the parameters of what is claimed. That is simply not logically possible here: the terms "content dependent data decompression" and "content independent data decompression" have no meaning and the concepts therefore contain no limitations that are understandable.

For the reasons set forth below, the Court grants defendants' motion for partial summary judgment in its entirety.

BACKGROUND

I. The Claims At Issue

At issue on this motion are two independent claims of the '747 Patent (claims 1 and 8) and one independent claim of the '651 Patent (claim 1).[3] Defendants refer to claim 1 of the '747 Patent as "representative" of the other claims--and this Court finds that it is. That claims states:

> 1. A method of decompressing one or more compressed data packets of a data stream using a data processor, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to a data packet, the method comprising:
>
> Receiving a data packet from the data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders used to compress data blocks associated with the data packet, and further wherein the lossless encoders are selected based on analyses of content of the data blocks;

[3] The remainder of the claims that are part of this motion are dependent claims.

> analyzing the data packet to identify a descriptor;
>
> selecting one or more lossless decoders for a data block associated with the data packet, wherein the selecting is based on the descriptor;
>
> decompressing the data block with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data block is encoded utilizing content dependent data compression; and
>
> decompressing the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression.

'747 Patent, claim 1 (Decl. of Nicole E. Feit in Support of Defs.' Mot. for Partial Summ. J. ("Feit Decl.") (Dkt. No. 430) Ex. A).

II. <u>The Invention</u>

The clearest way to understand the correctness of defendants' position is to understand the inventions set forth in the '747 and '651 Patents.

The '747 Patent is entitled "Data **Compression** Systems and Methods." (Feit Decl. Ex. A (emphasis added).) The Summary of the Invention describes "systems and methods for providing fast and efficient data compression using a combination of content independent data compression and content dependent data compression." No concept of content dependent or independent **decompression** is mentioned anywhere in the "Summary" section. That is unsurprising since the invention is concerned with a

number of steps that attempt to improve the speed and efficiency of compression.

The '747 Patent describes techniques in which content dependent data compression is performed on a data block when the data type (such as ACSII, MPEG, etc.) is identified; when the data type is not identified, then content independent data compression is utilized. In both cases, in connection with analysis of the content of the data block, the appropriate lossless compression algorithm(s) is selected and applied. The patent is clear that lossless compression algorithms are well known in the art and the invention is not creating a new lossless compression algorithm.

In addition, a descriptor containing information regarding which compression algorithm(s) was used is appended to the data block. The descriptor plays an important role in the invention, it "indicate(s) the data encoding technique applied to the encoded data block." (Patent '747 at 14:17-20.) The data block then travels to the decoder module. The decoder module includes a plurality of decoders for decoding the input data block using a decoder, set of decoders, or sequential set of decoders corresponding to the extracted compression type descriptor. The decoders include those "lossless encoding techniques currently well-known within the art." (Patent '747 at 14:60-65.)

In short, the data block is analyzed, if its data type is determined, content dependent data compression is utilized and a descriptor appended indicating which compression algorithm was used; if its data type is not recognized, then content independent data compression is utilized, and a descriptor is appended indicating which algorithm(s) was used to compress the data type(s). The descriptor is then read on the other end, during data decompression, and the appropriate matching algorithm is applied. Once the algorithm(s) has been determined and noted in the descriptor, whether compression was done "dependently" or "independently" of the content loses its relevance. More to the point, all that is left to be done in decompression is to reverse the compression (however that compression was achieved) using the matching algorithm.

DISCUSSION

I. Legal Principles

A. Summary Judgment

Summary judgment is appropriate when there is no genuine issue of fact requiring a trial, and such absence demonstrates that the movant is entitled to judgment as a matter of law. See Fed. R. Civ. P. 56(a); see also Celotex Corp. v. Catrett, 477 U.S. 317, 322 (1986). Conclusory or speculative assertions are insufficient to create a material issue of fact. Fletcher v. Atex, Inc., 68 F.3d 1451, 1456 (2d Cir. 1995).

Issued patents carry a presumption of validity. 35 U.S.C. § 282. A party seeking to invalidate a patent at summary judgment must submit "clear and convincing evidence of facts underlying invalidity" such that no reasonable fact finder could find otherwise. See Trimed, Inc. v. Stryker Corp., 608 F.3d 1333, 1340 (Fed. Cir. 2010); Laryngeal mask Co. Ltd. V. Ambu A/S, 618 F.3d 1367 (Fed. Cir. 2010).

B. 35 U.S.C. § 112

35 U.S.C. § 112 provides, in pertinent part, that a patent specification

> shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor of carrying out his invention.
>
> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

35 U.S.C. § 112 ¶¶ 1-2.

In determining whether a specification is sufficiently definite, a court must determine whether those skilled in the art would understand the scope of the claim when the claim is read in light of the remainder of the specification to which it relates. See Union Pac. Res. Co. v. Chesapeake Energy Corp., 236 F.3d 684, 692 (Fed. Cir. 2001). The claims must "delineate

the scope of the invention using language that adequately notifies the public of the patentee's right to exclude." Datamize, LLC v. Plumtree Software, Inc., 417 F.3d 1342, 1347 (Fed. Cir. 2005).

There are a number of cases in which summary judgment has been granted for failure to comply with the written description requirement of 35 U.S.C. § 112. See, e.g., Noah Sys., Inc. v. Intuit Inc., 675 F.3d 1302 (Fed. Cir. 2012); Atl. Research Marketing Sys., Inc. v. Troy, 659 F.3d 1345, 1353 (Fed. Cir. 2011); Athletic Alternatives, Inc. v. Prince Mfg., Inc., 73 F.3d 1573, 1581 (Fed. Cir. 1996).

Determining whether a patent claim fails to meet the definiteness and written description requirements of section 112 requires both legal and factual determinations. See IGT v. Bally Gaming Int'l, Inc., 659 F.3d 1109, 1119 (Fed. Cir. 2011). Courts cannot declare a claim indefinite simply because it may be difficult to construe: a claim is definite if its meaning is discernible, even though the effort of arriving at that definition may be formidable. See id. ("A claim is definite if one skilled in the art would understand the bounds of the claim when read in light of the specification. A claim is only indefinite if it is not amenable to construction or is insolubly ambiguous."); Bancorp Servs., LLC v. Hartford Life Ins. Co., 359 F.3d 1367, 1372 (Fed. Cir. 2004). However, a claim will be

deemed insolubly ambiguous where a movant can show by clear and convincing evidence that one skilled in the art cannot discern the outer boundaries of the claim based on the language of the claim, the specification, the prosecution history, and knowledge of the relevant art. Halliburton Energy Servs., Inc. v. M-I LLC, 514 F.3d 1244, 1249 (Fed. Cir. 2008).

II. Analysis

Here, the question is whether anyone can understand what the terms "content dependent" and "content independent" data decompression mean--that is, how is data decompression content dependent or independent and/or what techniques described in the invention relate to or rely upon those concepts? What would it mean to have decompression be content dependent or content independent? Those are questions which must be comprehensible to one skilled in the art in order for the claims at issue to pass muster under 35 U.S.C. § 112.

Defendants argue that the claims at issue fail to meet the definiteness and written description requirements of 35 U.S.C. § 112 for the following reasons: first, defendants assert that the terms "content dependent data decompression" and "content independent data decompression" are indefinite because, put simply, nothing differentiates "content dependent" from "content independent" data decompression. (Defs. Mem. of Law in Supp. of Mot. for Partial Summ. J. ("Defs. Mem.") (Dkt. No. 429) at 11.)

Stated otherwise: data decompression uses the algorithms chosen in connection with the encoding process--i.e., the words "content independent" or "content dependent" have no meaning at the decompression stage of the process.

Thus, according to defendants, it is "impossible to determine whether any particular data decompression technique counts as "'content dependent data decompression,' 'content independent data decompression,' both or neither." (Defs. Mem. at 11.) Thus, according to defendants, the terms are insolubly ambiguous. Defendants' argument boils down to the proposition that each limitation of a claim must be construed so that it has meaning; here, they argue that that cannot be done.

Second, defendants argue that even if the terms "content dependent" and "content independent" data decompression are not insolubly ambiguous (which the Court finds they are), the claims to which they relate nonetheless fail the written description requirements of 35 U.S.C. § 112. Defendants claim that this requirement has not been met because (a) the term "content dependent data decompression" is never used in the specification of either patent and "nothing in either patent provides either term with any written description support for any definite or substantive meaning," and (b) there is no written description support for the limitations that require decompression using either content dependent or content independent techniques.

(Defs. Mem. at 21.) All of defendants' arguments turn on the same fundamental proposition (explored further below) that no part of the data decompression process requires--or even involves--reference to the data content.

There are a number of steps in the data compression process that refer to aspects of the invention that require data analysis to determine if content dependent or content independent data compression is utilized. However, as mentioned above, the specification is perfectly clear on its face--and it is likewise clear in the declarations of both experts submitted in connection with this motion--that once the algorithm(s) has been determined and noted in the descriptor, whether compression was done "dependently" or "independently" of the data content loses its relevance. More to the point, all that is left to be done in decompression is to reverse the compression (however that compression was achieved) using the matching algorithm. On this, both experts agree. (Storer Decl. ¶¶ 22, 24, 32; Shamos Decl. ¶ 17.) There is no step in the actual decompression process that requires content analysis or reference--content independence and dependence have no meaning during data decompression. Thus, the usage of the terms "content" and "dependent/independent" in relation to decompression make no sense: they are insolubly ambiguous.

In support of plaintiff's opposition to this motion, Dr. Shamos makes a number of statements directly supportive of defendants' positions. He states that "the descriptor tells the system how decompression should be performed." (Shamos Decl. ¶ 8 (emphasis added).) He conceded that never once in the "Summary of Invention" is decompression mentioned. (Id. ¶ 9.) He states, "All that is necessary (e.g., in claim 1 of the '651 Patent) is for the decoding process to select the proper decoder corresponding to the encoder that was used to compress the data being decompressed." (Id. ¶ 14.) He also states that "[a]dding 'de' to 'compression' to yield the word 'decompression' simply means reversing the effect of compression." (Id. ¶ 15.) He further states that "the decompression algorithm is the reverse of whatever compression algorithm was used to encode the data." (Id. ¶ 17.)[4]

Dr. Shamos' declaration provides no support for any step in the data decompression process that requires content analysis--

[4] Dr. Shamos attempts to illustrate defendants' incorrectness by using an example of two types of hotel unlocking methods: key and wireless. He incorrectly states that defendants would have it "that there is no discernible difference between key unlocking and wireless unlocking (although there clearly is a difference) because in both of them the unlocking method is the reverse [of] whatever method was used to perform locking." (Shamos Decl. ¶ 17.) That misunderstands defendants' point. The point is that once it is determined at the initial stage whether key or wireless unlocks the door (e.g., the equivalent of determining which algorithm to use during data compression), then all one needs to do is discard the useless technique, for instance holding onto the key, and use the key at the back end. Once one knew that the key unlocks the door and not a wireless technique, it would certainly make no sense to try to use the wireless technique. So too here: once the content analysis has occurred once, and an algorithm has been chosen, there is no need to, and the invention never requires that, the content be analyzed in connection with data decompression.

or what content dependency or independency in connection with data decompression could mean.[5] This necessarily means that the terms "dependent" and "independent" data decompression are without meaning--or if they have meaning, they are insolubly ambiguous in the manner in which the invention is described and set forth. This is consistent with the declaration of Dr. Storer, submitted in support of defendants' motion.[6]

Dr. Storer states, and Dr. Shamos does not dispute, that both the '747 and '651 Patents are principally focused on data compression. (Storer Decl. ¶ 18.) Dr. Storer states, and Dr. Shamos does not contradict, that "the terms 'content dependent data compression' and 'content independent data compression' are not terms of art used in the data compression field." (Id. ¶ 31.) According to Dr. Storer, "'Content independent data decompression' or 'content dependent data decompression' are not terms with known meanings to persons of skill in the art." (Id. ¶ 34.)[7] He also confirms, as Dr. Shamos asserted, that

[5] He simply makes the conclusory assertion that "'content dependent data decompression' simply means a process that reverses the effect of 'content dependent data compression.' 'Content independent data decompression likewise means a process that reverses the effect of 'content independent data compression.'" (Shamos Decl. ¶ 7.) That is insufficient to create a triable issue of fact. See Fletcher, 68 F.3d at 1456.

[6] Dr. Storer is the founder of the Annual Data Compression Conference, he has researched and written extensively in the areas of data compression and has authored or coauthored over one hundred publications in the areas of data compression, content based image retrieval, image and video processing and parallel computing. He has written a textbook on data compression. (Storer Decl. ¶¶ 10, 12.)

[7] Dr. Shamos does not outwardly contradict that assertion, but instead states that "[a]ll that is needed is an understanding of whether the corresponding

decompression in the '747 and '651 Patents is the reverse of the compression algorithm used to encode the data: "An encoded data block is decoded in a way that Realtime calls the 'reverse' of the encoding algorithm. If decompression is attempted using a different algorithm, the decompressed data is incomprehensible and useless." (Id. ¶ 32; see also Shamos Decl. ¶¶ 15, 17.) Regarding the role of the descriptor, Dr. Storer asserts, "The patent teaches that the descriptor appended to the data block will identify which algorithm was used," but adds that there is nothing in the specification which indicates that the descriptor would contain any information as to whether the encoding was content dependent or content independent. (Storer Decl. ¶ 33.) Dr. Shamos states that the descriptor "tells the system how decompression should be performed" because "if the system does not understand the significance of the descriptors in the encoded data stream, it cannot know which decompressor to use on a particular data field or data block and decompression will fail." (Shamos Decl. ¶ 8.) But what Dr. Shamos omits from that analysis is precisely the step that Dr. Storer discusses--i.e., the descriptor providing information for the choice of

---

encoder was chosen by knowing the data block type (content data decompression) or not knowing the data block type (content independent data compression)." (Shamos Decl. ¶ 16.) That is simply a regurgitation of Dr. Shamos' "reverse" argument that the Court finds not only conclusory, but also not helpful to illuminating how decompression could be content dependent or independent. That is likely because decompression is, as discussed, not. Rather, decompression depends simply upon the algorithm used at the time of compression.

algorithm. As discussed at length, once the algorithm is selected during compression, all that needs to be done is apply the reverse of that algorithm during decompression. (See, e.g. Shamos Decl. ¶¶ 15, 17.)

Dr. Storer states that content dependent and content independent data compression are techniques for finding the right algorithm to use--they are not themselves algorithms. (Storer Decl. ¶ 36.) That means that once the **technique** for compression is used (e.g., dependent or independent), and information about the algorithm selected as a result of that process is placed in the descriptor, there is no logical reason, and none is explained, as to what content dependent or independent data decompression **could** mean. As Dr. Storer states, and this is in fact supported by Dr. Shamos,

> once the data has been encoded, a user skilled in the art has essentially no choice as to what algorithm is employed to decompress. It has long been known in the field that data compression can be 'reversed' using the same (or the reverse of the) algorithm that was used to compress it. Otherwise, there is no useful output.

(Id. ¶ 41.) Dr. Storer states, and this is not contradicted by Dr. Shamos (whose references to understanding the decompression algorithm are focused on the information in the descriptor), that "the patents do not know how to choose that decompression algorithm based on whether 'content dependent' or 'content independent' data compression had been 'utilized' to select the

compression algorithm that provided the highest level of compression." (Id.)

However, as Dr. Storer points out, despite the insoluble ambiguity in the terms when used in connection with decompression, they nonetheless purport in the claims to impose limitations. Indeed, they are set forth in the claims as mutually exclusive alternatives. Specifically, claim 1 of the '747 Patent requires that

> Decompressing the data block with a selected lossless decoder utilizing content dependent data decompression if the descriptor indicates the data block is encoded utilizing content independent data compression, and
>
> Decompressing the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression.

'747 Patent, claim 1.

As Dr. Storer notes, and this is similarly uncontradicted by Dr. Shamos, decompression is based on the compression algorithm--not the technique used to choose that algorithm. (See, e.g., Storer Decl. ¶¶ 41-43.) Dr. Storer concludes, "a person skilled in the art would not be able to understand the meaning of the patent claims because they provide no discernible distinction between 'content dependent data decompression' and 'content independent data decompression.'" (Id. ¶ 46.) He correctly notes that since Dr. Shamos conceded that

decompression is merely reversing the method of compression, then the words in the claim set forth above following the word "utilizing" are superfluous. (Id. ¶ 47.) In other words, the claim limitation could read, without any loss of meaning, "decompressing the data block with a selected lossless decoder."

In addition to the complementary descriptions from both experts as to what occurs as part of the decompression process, there is additional support for "content dependent data decompression" failing the written description requirement. The attorney who wrote the applications for the '747 and '651 Patents, as well as for a lineal ancestor of both (i.e., the '024 Patent), testified at his deposition that the compression methods and system disclosed in the '024 Patent had nothing to do with content dependent data compression or decompression. (Defs.' Supplemental Submission in Support of Partial Summ. J. ("Defs. Supp.") (under seal) at 3.) The systems and methods of the '024 Patent provide material with respect to only content independent data compression; thus, it is not possible that the '024 specification provides any basis for a content dependent claim or limitation. (Id.) The patent attorney testified that no material was added to the continuation in part of the '024 Patent in connection with the other patent applications that added material relating to content dependency. (Id.)

Moreover, Dr. Shamos has stated that the conventional decompression methods as disclosed in the '024 patent "would fail at least because they would not be able to make use of the descriptors in the data stream." (Shamos Decl. ¶ 6.) Thus, the patent specifications do not have written material that tells the public anything about specific decoders that would be used in connection with content independent data decompression or content dependent data decompression. (Defs. Supp. at 4.)

There is no factual dispute between Drs. Storer and Shamos that creates a triable issue on this motion. The terms "content dependent data decompression" and "content independent data decompression" are undefined and meaningless. Dr. Shamos' declaration does not provide any meaning to the terms. While content dependency and independency have a great deal to do with the compression claims in the inventions, they lose all meaning once the encoding process has occurred and the descriptor is appended. All that matters from that point forward is what encoder was used--not the method of its selection (i.e., not the content on which the encoder selection was based). Simply put, decompression has everything to do with the algorithm used at the front-end compression and nothing to do with the content on which the selection of that algorithm was based. That presents a mind-twisting conundrum regarding the meaning of the terms at issue--i.e., a conundrum that 35 U.S.C. § 112 was designed to

prevent. Accordingly, the terms cannot be construed, they are indefinite, and the claims purporting to be based upon them are invalid.

In addition, however, use of the terms in the claims fails the written description requirement. There is no guidance in the specifications or claims, and none that either expert was able to provide on this motion, that would put members of the public on notice as to what is meant to be captured by content dependent data decompression that is distinct from content independent data decompression that is distinct from routine reversal of compression by use of well known methods of decompression. 35 U.S.C. § 112 requires more. Accordingly, the claims at issue fail the written description requirement.

CONCLUSION

For the reasons set forth above, defendants' motion for partial summary judgment is GRANTED.

Claims 1, 7, 8 and 13 of the '747 Patent and Claims 1, 4, 6, 7 and 12 of the '651 Patent fail to comply with the definiteness and written description requirements of 25 U.S.C. § 112 and thus, are invalid.

The Clerk of the Court is directed to terminate the motion at Docket No. 428 in 11 Civ. 6696.

SO ORDERED:

Dated: New York, New York
June 27, 2012

K B. Forrest

KATHERINE B. FORREST
United States District Judge

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- MORGAN STANLEY, et al., Defendants.	11 Civ. 6696 (KBF) 11 Civ. 6701 (KBF) 11 Civ. 6704 (KBF) MEMORANDUM & ORDER
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- CME GROUP INC., et al. Defendants.	11 Civ. 6697 (KBF) 11 Civ. 6699 (KBF) 11 Civ. 6702 (KBF) MEMORANDUM & ORDER
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- THOMSON REUTERS, et al. Defendants.	11 Civ. 6698 (KBF) 11 Civ. 6700 (KBF) 11 Civ. 6703 (KBF) MEMORANDUM & ORDER

KATHERINE B. FORREST, District Judge:

Plaintiff Realtime Data, LLC ("Realtime" or "plaintiff") has sued over 18 defendants in nine separate actions for patent infringement.[1] The actions have been consolidated for pre-trial

[1] The number of defendants varies according to whether parent, subsidiaries and sister companies are grouped together or separately. If counted as individual defendants, the total number of defendants is over 30; grouped according to family of company the total is 18.

proceedings. Realtime served initial sets of infringement contentions in 2010 (the "2010 infringement contentions") and then supplemental infringement contentions at various points during the spring of 2012 (the "supplemental infringement contentions"). In its 2010 infringement contentions, Realtime asserted direct infringement and included a "placeholder" for the doctrine of equivalents. Two years later, its supplemental infringement contentions did not contain a reference to the doctrine of equivalents.

On June 15, 2012, Realtime submitted an expert report from Michael Ian Shamos, Ph.D. (the "Shamos Report") relating to infringement of the patents-in-suit. The Shamos Report sets forth opinions stating that defendants have infringed the patents directly and under the doctrine of equivalents. Defendants have moved to strike those portions of the Shamos Report that contain opinions regarding the doctrine of equivalents.

For the reasons set forth below, defendants' motion to strike is GRANTED.

DISCUSSION

The procedural history of this matter is complex and each of the dockets relating to the various above-captioned actions now amount to entries covering more than 100 pages. For

purposes of this motion, the Court refers only to the procedural facts necessary to resolution of this motion.

The story of the infringement contentions started more than two years ago in the United States District Court for the Eastern District of Texas, where certain defendants in the above-captioned actions were originally sued. Those actions (the "Texas actions") were later transferred to this Court. During the time that those actions were pending in the Eastern District of Texas, the parties were following that court's Local Patent Rules. See Local Rules, E.D. Tex., Appx. M ("Patent Rules") (July 16, 2012), available at http://www.txed.uscourts.gov/page1.shtml?location=rules. Among the relevant patent rules was a requirement that "not later than 10 days" prior to the case management conference, "a party claiming infringement must serve on all parties a 'Disclosure of Asserted Claims and Infringement Contentions." Id. at Rule 3-1 (emphasis added). That disclosure was required to include, inter alia, "whether each element of each asserted claim is claimed literally present or present under the doctrine of equivalents." Id. at 3-1(d).

In the Texas actions, Realtime served an initial series of infringement contentions on January 22, 2010. Realtime served additional infringement contentions on July 2, 2010 and October 15, 2010 (what the Court has defined as the "2010 infringement

contentions"). Realtime's 2010 infringement contentions contained boilerplate references to the possibility of future assertions under the doctrine of equivalents. For instance, in its disclosure to CME Group., Inc., et al., Realtime made the following general statements:

> For any limitation that is found to be not literally present, Realtime Data asserts that defendants induce and/or contribute to infringement by others and/or that such limitation is present under the doctrine of equivalents.
>
> At present, Realtime Data lacks specific knowledge as to which, if any, limitations of the asserted claims the named defendants believe not to be literally embodied by the accused products and, hence, as to whether Realtime Data will be contending that any limitations of the asserted claims (and if so, which ones) are equivalently embodied by the accused products. To the extent that any differences are alleged to exist between the above-identified claims and the Accused Instrumentalities/acts, such differences are insubstantial. Defendants' Accused Instrumentalities perform substantially the same function, in substantially the same way to yield substantially the same result, and therefore Defendants infringe under the doctrine of equivalents. Moreover, pursuant to Patent Local Rule 3-6, Realtime Data reserves the right to amend its Infringement Contentions to specifically assert infringement by the doctrine of equivalents in light of the Court's claim construction.

Following transfer to this Court, a schedule was established for all remaining pre-trial proceedings and a firm trial date was set for November 26, 2012. When this Court set that schedule, Realtime did not state that it would need an

additional opportunity to amend its contentions to add detail regarding the doctrine of equivalents.

At a conference with the Court on March 16, 2012, Realtime represented that it had served supplemental infringement contentions the prior Monday. (Tr. of Conf. (Mar. 16, 2012) (hereinafter "Tr.") at 39.) At that conference, Realtime stated that it needed to complete its review of source code and would then provide defendants with its final infringement contention "maps" (or charts) by April 16, 2012. (Tr. at 44.) Defendants committed to serve final invalidity contentions by April 30, 2012. Realtime did not ask the Court for an opportunity to amend its infringement contentions in light of the parties' proposals on claim construction or to have an opportunity to revise those contentions following a ruling on claim construction.

Realtime served many defendants with supplemental infringement contentions in March 2012 (and continued to supplement its contentions subsequent to that date). Realtime's supplemental infringement contentions contain no boilerplate reservations and do not mention the doctrine of equivalents at all. In its letter to this Court opposing defendants' motion to strike, Realtime argues that the supplemental infringement contentions were just that--supplemental. (See Letter from Robert A. Cote, Counsel for Realtime, to Hon. Katherine B.

Forrest (June 29, 2012) ("Realtime June 29 Letter") at 2 ("Defendants suggest that because supplemental claim charts (not an Amended 3-1(d) Disclosure, a substantive distinction, do not themselves explicitly state reliance on the DOE, Realtime has somehow waived its reliance. To the contrary, Realtime has never expressly or impliedly disclaimed reliance on the DOE.").)

The supplemental infringement contentions contained substantial additions to, and ultimately modifications of, the 2010 infringement contentions: claims of infringement as to certain patents were dropped altogether, and for some defendants the Accused Instrumentalities changed entirely. In short, for a number of defendants the references in the Initial Infringement Contentions to Accused Instrumentalities became entirely irrelevant since those instrumentalities were no longer at issue.

A. The Claim Construction Process

As discussed further below, plaintiff's assertions regarding the doctrine of equivalents are based on the parties' proposed alternative claim constructions. A chronology of events regarding the claim construction process thus provides useful context.

In February 2012, the parties exchanged proposed claim constructions. Opening briefs on terms requiring construction were due on March 16, 2010. Between February and March, 2012,

Realtime knew the differences between the parties' proposed constructions; if it believed that such differences were essentially meaningless, it could have included specific contentions regarding the doctrine of equivalents in the infringement supplements it served in March 2012. If for some other reason it believed claim construction raised doctrine of equivalents issues, it should have immediately raised that issue with the Court. It failed to do so.[2]

The Court held a Markman hearing on May 4, 2012. At that hearing, the parties presented live expert testimony. The Court posed a number of questions. Throughout the proceeding, neither counsel for plaintiff nor plaintiff's expert ever indicated that certain proposed constructions for particular terms were substantially the same--indeed, there was extensive testimony and briefing on the differences in meaning and effect of the proposed constructions.

On June 15, 2012, plaintiffs provided defendants with the Shamos Report in which the doctrine of equivalents features prominently. In that report, Dr. Shamos opines that whether the

[2] Realtime argues that its March supplements were merely claim charts--and that there is no requirement (in this Court's Individual Patent Rules or the Eastern District of Texas' local patent rules) that claim charts contain details on the doctrine of equivalents. (See Realtime June 29 Letter at 2-3.) First, that ignores that these documents are, on their faces, entitled "Supplemental Infringement Contentions." The argument also ignores the plain fact that nowhere--not in the claim charts and not in any supplemental or amended contentions--did Realtime ever set forth the required detail regarding its assertions that defendants have infringed under the doctrine of equivalents.

Court ultimately adopted plaintiff's claim construction or defendants' would be irrelevant--since defendants nevertheless infringed under the alternative construction pursuant to the doctrine of equivalents.

The Court issued its Markman ruling on June 22, 2012. The Court in large part construed the claims consistently with defendants' proposed constructions.

B. The Shamos Report

Plaintiff never amended its infringement contentions to add any of the required detail regarding the doctrine of equivalents. This Court's Local Patent Rule 1(a)(v) requires that "[n]ot later than 14 days after appearance in th[e] action," a party must disclose "[w]hether each limitation of each asserted claim is alleged to be literally present or present under the doctrine of equivalents in the Accused Instrumentality." Individual Practices in Civil Cases, Judge Katherine B. Forrest, at Appx. B at Rule 1(a)(v). Plaintiff therefore failed to preserve infringement claims under the doctrine of equivalents. See Genentech, Inc. v. Amgen, Inc., 289 F.3d 761, 774 (Fed. Cir. 2004)(failure to give proper notice to alleged infringer of doctrine of equivalents barred use of the doctrine); Optimumpath, LLC v. Belkin Int'l, Inc., 2011 WL 1399257, at *8 (N.D. Cal. Apr. 12, 2011)(recognizing that attempts to assert claims under the doctrine of equivalents with

blanket statements have been rejected); Rambus Inc. v. Hynix Semiconductor, Inc., Nos. C-05-00334, C-05-2298, C-06-00244, 2008 WL 5411564 (N.D. Cal. Dec. 29, 2008)(boilerplate reservations insufficient); Nike, Inc. v. Adidas Am., Inc., 479 F. Supp. 2d 664, 670 (E.D. Tex. 2007) (Nike's initial infringement contentions contained two boilerplate paragraphs regarding the doctrine of equivalents; later contentions do not mention the doctrine; the court declined to allow assertion of the doctrine when discovery was to close in a month); MEMC Elec. Materials v. Mitsubishi Materials Silicon Corp., No. C 01-4925, 2004 WL 5363616, at *5 (N.D. Cal. Mar. 2, 2004)(a blanket statement that does not identify where each element of each claim is found is insufficient and does not point out each asserted element of each asserted claim that there is infringement of the doctrine of equivalents).

The Shamos Report does, however, opine extensively as to infringement under the doctrine of equivalents. The Shamos Report is written in the nature of plaintiff "hedging its bets" as to the Court's Markman ruling (which was released one week later). In a sort of "heads I win, tails you lose" scenario, the Shamos Report finds infringement under either plaintiff's or defendants' proposed constructions.

The Shamos Report, however, is not a set of infringement contentions as contemplated by either Rule 1(a)(v) of this

Court's Individual Patent Rules or Rule 3-1(d) of the Eastern District of Texas' Local Patent Rules. There is no provision in this Court's Individual Patent Rules for infringement contentions to be revealed for the first time by way of expert report, or at such a late date.

C. Plaintiff Argues No Prejudice

Plaintiff argues that the Shamos Report does not result in prejudice to any defendant.

As an initial matter and as discussed above, it is clear that plaintiff has not complied with relevant local or individual patent rules regarding the detail required for infringement contentions. Plaintiff cannot now, sua sponte, amend those rules or give itself a "pass" as to those rules without having first sought leave of court--and no such leave was sought. This Court's Individual Patent Rules are necessary to insure a fair and orderly resolution of a patent dispute. This particular matter has required substantial work by this Court to manage--indeed, the Court has analogized it to the "herding of cats" and indeed, many, many cats of various varieties. Nevertheless, the matter is on track for trial in November and this Court has every expectation that it will proceed as scheduled. Maintaining orderly case management is necessary to resolution of this matter and the effective and efficient administration of justice. Compliance with this

Court's Individual Patent Rules is part of that process. Plaintiff simply failed to comply with those rules and cannot do so now through a "backdoor process." Several defendants have stated that they will, in fact, be significantly prejudiced by belated assertions of infringement under the doctrine of equivalents.[3] Defendant Thomson Reuters has stated that it would have developed a factual record regarding Dr. Shamos' theories but is now unable to do so because discovery has closed. (See Letter from Constance S. Hutner, Counsel for Thomson Reuters to Hon. Katherine B. Forrest (July 17, 2012) at 2 ("Realtime's unexcused delay in raising its DOE allegations has limited Defendants' ability to identify prior art that would render the patents-in-suit invalid. If Defendants had notice of Realtime's theories of equivalents during fact discovery, they would have focused on prior art falling within the alleged range of equivalents…Having such knowledge might also have impacted the third party depositions Defendants chose to take and Defendants' invalidity positions . . . [it] has also prevented Defendants from pursuing fact discovery concerning the insubstantial differences test."); see also Letter from Constance S. Hutner to Hon. Katherine B. Forrest (June 28, 2012) at 3.)

[3] As the Court noted, fact discovery in this matter closed prior to defendants' receipt of the Shamos Report.

Similarly, NYSE Euronext, NYSE Arca, Inc., NYSE Amex, LLC, Securities Information Company (collectively "NYSE"), and Options Price Reporting Authority LLC ("OPRA"), stated that their summary judgment motions (now submitted according to a schedule previously set), Markman positions, discovery and invalidity positions "have all been developed in response to the infringement contentions Realtime actually stated." (Letter from James H. Shalek to Hon. Katherine B. Forrest (June 29, 2012) at 2.)

Defendants Morgan Stanley, Morgan Stanley & Co. Inc., Goldman Sachs Group, Inc., Goldman Sachs & Co.,, Goldman Sachs Execution & Clearing, L.P., J.P. Morgan Chase & Co., J.P. Morgan Securities, Inc., and J.P. Morgan Clearing Corp. (collectively referred to as the "Bank Defendants"), also stated:

> The prejudice stemming from Realtime's failure to provide proper notice of its infringement theories under the doctrine of equivalents is neither theoretical nor insubstantial. Indeed, our entire case strategy has been developed under the understanding that Realtime was not pursuing claims based on the doctrine of equivalents.

(Letter from Daniel A. DeVito to Hon. Katherine B. Forrest (June 29, 2012) at 2.) The Bank Defendants also assert that they would have developed additional facts as part of the discovery process, now foreclosed. (Id.)

Had defendants understood that plaintiff would assert that different claim constructions would result in "insubstantial

differences" regarding infringement, the discovery process would have taken that position into account. This case has been too thoroughly and actively litigated for that not to have been the case. There is undoubted prejudice to defendants now to allow only a battle of the experts without the development of a factual record on this issue. That is not the way litigation is supposed to work and it will not work so here.

CONCLUSION

For the reasons set forth above, this Court GRANTS defendants' motion to strike those portions of the Shamos Report and any related testimony regarding the doctrine of equivalents. The letters underlying this issue will be filed publicly on the docket unless the parties notify the Court within five (5) days that they should be filed under seal.

SO ORDERED:

Dated: New York, New York
August 2, 2012

K. B. Forrest

KATHERINE B. FORREST
United States District Judge

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- MORGAN STANLEY, et al., Defendants.	11 Civ. 6696 (KBF) 11 Civ. 6701 (KBF) 11 Civ. 6704 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- CME GROUP INC., et al. Defendants.	11 Civ. 6697 (KBF) 11 Civ. 6699 (KBF) 11 Civ. 6702 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- THOMSON REUTERS, et al. Defendants.	11 Civ. 6698 (KBF) 11 Civ. 6700 (KBF) 11 Civ. 6703 (KBF) OPINION & ORDER

KATHERINE B. FORREST, District Judge:

In 2009, plaintiff Realtime Data, LLC d/b/a IXO ("Realtime") sued a number of companies involved in some aspect of the financial services industry (including banks, exchanges, and information services) for infringing three of its patents: U.S. Patent No. 7,417,568 (the "'568 Patent"), U.S. Patent No. 7,714,747 (the "'747 Patent"), and U.S. Patent No. 7,777,651

(the "'651 Patent" and collectively with the '568 and '747 Patents, the "patents-in-suit").

This Court has already issued several decisions on various issues in this litigation, and refers to those decisions for additional facts. See, e.g., Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 3158196 (S.D.N.Y. Aug. 2, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 2545096 (S.D.N.Y. June 27, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 2434750 (S.D.N.Y. June 26, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 2394433 (S.D.N.Y. June 22, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 1711117 (S.D.N.Y. May 10, 2012).

Defendants have counterclaimed for non-infringement and invalidity. On July 19, 2012, the Court ordered that the trial as to the Exchange Defendants will proceed first.[1] (See 11 Civ.

[1] The "Exchange Defendants" are NYSE Euronext, NYSE ARCA, Inc., NYSE AMEX, LLC, Securities Industry Automation Corporation, Options Price Reporting Authority, LLC, International Securities Exchange, Boston Options Exchange Group LLC, CME Group Inc., Board of Trade of the City of Chicago, Inc., New York Mercantile Exchange, Inc., Chicago Board Options Exchange, Incorporated, BATS Trading, Inc., NASDAQ OMX Group, Inc., and NASDAQ OMX PHLX, Inc.

The term "Bank Defendants" refers to BNY ConvergEx Group LLC, BNY ConvergEx Execution Solutions LLC, Credit Suisse Holdings (USA), Inc., Credit Suisse Securities (USA) LLC, The Goldman Sachs Group, Inc., Goldman Sachs & Co., Goldman Sachs Execution & Clearing, L.P., HSBC Bank USA, N.A., HSBC Securities (USA), Inc., J.P. Morgan Chase & Co., J.P. Morgan Securities, Inc., J.P. Morgan Clearing Corp., Morgan Stanley, and Morgan Stanley & Co., Inc.

The term "Data Provider Defendants" refers to Thomson Reuters Corporation, Factset Research Systems, Inc., Bloomberg L.P., Interactive Data Corporation, Penson Worldwide, Inc., and Nexa Technologies, Inc.

6696, Dkt. No. 539.) That jury trial is scheduled to commence on November 26, 2012.

I. PROCEDURAL BACKGROUND

Defendants and plaintiff now have 19 separate, fully briefed motions for summary judgment (on the question of liability) pending before this Court.[2] The motions put forth a smorgasbord of separate arguments. Each motion asserts more than one basis for the proposition that as to one or more of the accused instrumentalities, Realtime cannot prove a required claim limitation. Accordingly, the 19 motions require the Court to evaluate the merits of far more than 19 arguments.[3]

Buried in one of those motions are issues amenable to resolution on summary judgment; however, many are not. It is important that a jury not be burdened with arguments that are properly resolved by the Court now. It is even more important that this Court not usurp the role of the jury by making determinations of fact, weighing evidence, or making credibility determinations.

[2] Although each of the motions was brought by a particular defendant (or group of defendants), a number of defendants have “joined” the motions of others. The Court notes such joinder in connection with the discussion of each motion addressed in this Opinion.

[3] Given the number of motions pending before the Court in these actions and given the pre-trial and trial schedule in the Exchange Action, for the sake of efficiency and expediency the Court has occasionally cited to the parties’ briefs for substantive points (which reference admissible evidence) rather than to the voluminous evidence (that is has reviewed and considered in connection with the pending motions).

Finding a path evenly distributed between judicial efficiency and the merits has been challenging. To assist the Court in sheer organization of the volume of paper, this Court assigned a number to each motion, and requested that the parties submit their views as to the order in which the Court should resolve the motions.

Based upon the Court's review of the letters submitted pursuant to that request (see 11 Civ. 6697, Dkt. Nos. 834 & 835), as well as its own review of the motions and the likelihood that resolving certain motions might assist in more rational trial preparation, this Court resolves the following motions in this Opinion: International Securities Exchange's ("ISE's") Motion for Summary Judgment of Noninfringement (Motion No. 4); CME Group Inc.'s ("CME") Motion for Summary Judgment of Noninfringement Based on the Lack of Determining a Data Block of Field Type (Motion No. 5); NYSE and Options Price Reporting Authority, LLC ("OPRA")'s Motion for Summary Judgment of Noninfringement Due to the Absence of the Encoding, Data Block (or Field) Type, and Selecting Limitations (Motion No. 6); NYSE, et al.'s Motion for Summary Judgment of Non-Infringement Descriptor Limitation (Motion No. 8); and Credit Suisse's Motion for Summary Judgment of Noninfringement: Descriptor Limitation

(Motion No. 9).[4] For ease of reference, the Court refers to the various motions by assigned number throughout this Opinion. When referring to all five motions collectively, the Court uses the term "Motions."[5]

II. THE PATENTS-IN-SUIT

Each Motion delves deeply into claim limitations of the patents-in-suit. Resolution of the Motions requires some understanding of the inventions at issue read against the backdrop of this Court's prior ruling on claim construction. See generally Realtime Data, LLC, 2012 WL 2394433 (referred to herein as the "Markman opinion").

Both the '651 and '568 Patents are entitled "System and Method for Data Feed Acceleration and Encryption." (See '568 Patent at [54]; '651 Patent at [54].) The '568 Patent issued first and discloses a system and methods for providing accelerated transmission of broadcast data, such as financial data and news feeds, over a communication channel using data compression and decompression to increase bandwidth and reduce latency. (See '568 Patent col.5 ll.25-32) The claims of the '568 Patent relate generally to methods for compressing data

[4] Resolution of the other pending motions for summary judgment will be resolved in subsequent opinions.

[5] The Court follows the order in which the parties set forth the arguments in their briefs, as well as the lettering system used by the parties for each of their disparate arguments in the respective Motions.

through encoding applied to a data stream. Certain claims relate to the method and application of selected encoders.

In contrast, the '651 Patent relates to a method of decoding one or more encoded messages. The decoding method requires receiving an encoded message, determining which decoder to utilize, and performing the decoding.

The '747 Patent claims both methods for decompressing one or more compressed data packets using multiple decoders that apply lossless decompression techniques, as well as methods for using multiple encoders that apply lossless compression techniques.

Key aspects of the inventions relate to, *inter alia*, how and when encoders and decoders are selected, how and when they are applied, to what they are applied, and whether they are in fact lossless.

Defendants universally claim that they have adopted an industry protocol referred to a compression and decompression standard called "FAST," and that FAST does not infringe on plaintiff's patents. Further, the "FIX Protocol" was copyrighted in 2006. It has been utilized by defendants.

III. GENERAL BACKGROUND

In connection with the Motions, there are certain facts relating to the FAST standard that are not materially in dispute.

A basic and overriding contention in this litigation is that FAST infringes the patents-in-suit. FAST is described in the Abstract of the FIX Protocol as "a space and processing efficient encoding method for message oriented data streams. It defines the layout of a binary representation and the semantics of a control structure called a template . . . ." (See Decl. of James Storer In Support of NYSE and OPRA Defs.' Mot. for Summ. J. of Noninfringement Based on the Absence of Encoding, Data Block (Or Field Type), and Selecting ("Storer Decl. Mot. 6") Ex. 1 at NYSE00083123.) According to the FIX Protocol, the FAST encoding method

> reduces the size of a data stream on two levels. First, a concept referred to as Field Operators allows the data affinities of a stream to be leveraged and redundant data to be removed. Second, serialization of the remaining data is accomplished through binary encoding which draws on self-describing field lengths and bit maps indicating the presence or absence of fields.

(Id. at NYSE00083128.)

FAST uses a template, or table, that identifies for each field the data type expected to be present in that field, and a field operator[6] that will be applied to the field during the encoding process. (See Decl. of James Storer in Support of Defs.' Mot. for Summ. J. of Non-Infringement: Descriptor Limitation ("Storer Decl. Mot. 8.") ¶ 11.) The template is not

[6] Realtime calls a "field operator" an encoder; defendants disagree that a field operator is an encoder. The Court need not resolve that issue in order to render its decision on the Motions.

attached to the data block that is compressed. Instead, it acts as a reference tool to identify the appropriate field operator. (See id. ¶ 19.) A template identifier (or "template ID") along with a presence map (or "PMAP") are used to reference the template and determine which field operator was used. (Id. ¶¶ 21-23.) The template ID does not itself state which field operator was used. (Id. ¶ 23.) The template is never transmitted with the compressed data from encoder to decoder. (Id. ¶ 21.)

IV. LEGAL STANDARD

Summary judgment is warranted if the pleadings, the discovery and disclosure materials, along with any (admissible) affidavits, demonstrate that there is no genuine issue of fact necessitating resolution at trial. Fed. R. Civ. P. 56(c); see also Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 247 (1986); Celotex Corp. v. Catrett, 477 U.S. 317, 322-323 (1986). A party moving for summary judgment bears the initial burden of demonstrating that no genuine issue of material fact exists; all reasonable inferences should be drawn in favor of the non-moving party. See Liberty Lobby, 477 U.S. at 255; Cont'l Can Co. USA, Inc. v. Monsanto Co., 948 F.2d 1264, 1265 (Fed. Cir. 1991). The burden then shifts to the non-moving party to come forward with "admissible evidence sufficient to raise a genuine issue of fact for trial in order to avoid summary judgment." Jaramillo v.

Weyerhauser Co., 536 F.3d 140, 145 (2d Cir. 2008); see also Glaverbel Societe Anonyme v. Northlake Mktg. & Supply, Inc., 45 F.3d 1550, 1560-61 (Fed. Cir. 1995) ("When the movant's burden of establishing the lack of a genuine issue of material fact has been met 'in facial terms,' the nonmovant must point to 'some evidence in the record sufficient to suggest that his view of the issue might be adopted by a reasonable factfinder.'" (quoting Resolution Trust Corp. v. Juergens, 965 F.2d 149, 151 (7th Cir. 1992))). Where the non-moving party would bear the ultimate burden of proof on an issue at trial, the moving party satisfies its burden on the motion by pointing to an absence of evidence to support an essential element of the non-movant's claim. See Intellicall, Inc. v. Phonometrics, Inc., 952 F.2d 1384, 1389 (Fed. Cir. 1992).

Where it is clear that no rational trier of fact could find in favor of the non-moving party, summary judgment is warranted. See Liberty Lobby, 477 U.S. at 248. However, the mere possibility that a dispute may exist, without more, is not sufficient to overcome a convincing presentation by the moving party. Id. at 247-48. Similarly, mere speculation or conjecture is insufficient to defeat a motion. W. World Ins. Co. v. Stack Oil, Inc., 922 F.2d 118, 121 (2d Cir. 1990).

In ruling on a motion for summary judgment, a court cannot, however, weigh the evidence or make credibility determinations:

those are the functions of the jury. See Liberty Lobby, 477 U.S. at 255. Further, when there are dueling experts, both of whom have put forward opinions in contradiction with each other, and when those opinions are important to resolution of a material factual dispute, summary judgment may not be appropriate. See Hodosh v. Block Drug Co., Inc., 786 F.2d 1136, 1143 (Fed. Cir. 1986) ("The fact issues herein must be resolved by trial in which the conflicting views of the experts will be subject to the refining fire of cross examination.").

The question is whether, at this stage of the proceeding, the court can determine whether the expert is merely asserting his own ipse dixit, which would be insufficient to defeat summary judgment, or whether two experts in the field could have reasonable differences. If it is the latter, then the Court must leave credibility determinations and the weighing of the experts' opinions to the jury.

V. MOTION 4[7]

ISE is accused of infringing both the '568 and '651 Patents, and has moved for summary judgment as to all claims of infringement asserted against it. It presents 11 arguments supported by an expert declaration from Dr. James A. Storer ("Storer Decl. Mot. 4") and a declaration from ISE's System and Product Strategy Officer, Gregory J. Maynard ("Maynard Decl.").

[7] No other defendants join in the motion.

In opposition, Realtime offers multiple responses to each of the 11 arguments as well as what it believes is support for its position(s) from the declaration of its own expert, Dr. Michael Ian Shamos ("Shamos Decl.").

According to Realtime, three of ISE's encoding products and methods are infringing instrumentalities: (1) ISE.Fast Encoding; (2) MIDAS Encoding; and (3) ISE.FastProcessing Encoding (collectively, the "ISE Accused Encoding Instrumentalities"). Realtime also accuses four of ISE's decoding products or methods of being infringing instrumentalities: (1) ISE.FastProcessing decoding; (2) Exergy OPRA Decoding; (3) OPRA Decoding; and (4) ISE.FastSpec Greeks Decoding (collectively, the "Accused Decoding Instrumentalities" and with the ISE Accused Encoding Instrumentalities, the "Accused Instrumentalities").

In order for any of the accused instrumentalities to infringe, it must satisfy the limitations in each asserted claim.

A. Do ISE's FAST Applications "Analyze" or "Recognize" to Determine Data Type?

Each of the claims that ISE is accused of infringing require "analyzing" or "recognizing" to determine data type. ISE argues that Realtime relies on a "value check" performed by the instrumentality at issue to determine the data block type of data field type, and that this fails to comply with this Court's Markman opinion which requires "content categorization," not

checking a value. (See Mem. of Law in Support of ISE's Mot. for Summ. J. of Noninfringement ("ISE Mem.") (11 Civ. 6697, Dkt. No. 685) at 3-4.)

According to ISE, checking the value of a particular field is not the same as "analyzing" it to determine data type. ISE refers to the following example: if the COPY operator is being applied to a particular field, and the previous message had a "5" associated with the same field, a program implementing FAST's COPY operator will check to see if the value of the current field is "5"--but it will not try to determine data type. (See ISE Mem. at 4.) ISE argues that the FAST data types are already known and are set forth in templates prior to compression.

Realtime responds that the encoder is not simply checking a value. According to Realtime, even in the example that ISE provides, ISE is checking the value against the value in a prior message. Realtime argues that this is sufficient to constitute an "analysis." (Realtime Data, LLC's Mem. in Opp'n to ISE's Mot. for Summ. J. of Non-Infringement ("Opp'n Mot. 4") at 5-6.)

There is no dispute that there is a comparison of values that occurs as part of ISE's encoding process. The issue is whether the two-step process constitutes an "analysis."[8] (Notably, as presented, that argument is not based on the type

[8] The Court notes that the parties did not seek construction of the term "analyze" in connection with encoding with respect to the Markman opinion.

of content to which the "analysis" relates.) The parties have submitted dueling expert opinions on this question. The Court cannot resolve the question of fact as to whether ISE's two-step process constitutes an "analysis." Accordingly, the Court rejects this argument as a basis for a finding of summary judgment, and denies this aspect of Motion No. 4.

B. Are Fast Field Operators Encoders?

ISE argues that its Accused Instrumentalities do not employ encoders. ISE correctly notes that this Court has previously construed the term "encoder" to mean "hardware or software that compresses data by converting the contents of a data block (or data field) into a coded representation of those contents." See Realtime Data, 2012 WL 2394433, at *16. The Court's construction makes clear that a key distinction is that encoding must be more than simply "compression"--it must include some form of "coding" or changing of representation. See id. at *8-9. The Court also determined that throwing data "away" or "no change" does not constitute "encoding." Id. at *9.

Realtime has asserted that three of ISE's accused instrumentalities use or include encoders: COPY, DEFAULT and INCREMENT. According to ISE, none of those instrumentalities use or include encoders because none "code" data. (See ISE Mem. at 6.) Realtime asserts that in fact all three of the accused instrumentalities generate a coded representation. (See, e.g.,

Shamos Decl. ¶¶ 16, 17.) It relies upon the fact that ISE products all use FAST encoding techniques, as stated by Dr. Storer. (See Opp'n Mot. 4 (citing Storer Decl. Mot. 4 at 5-8).) According to Realtime, lossless encoding techniques can include "field encoding"--and field encoding includes copy, default and increment encoding. (Id. at 7-8.) Realtime also argues that such encoding techniques are content dependent--i.e., that default encoding is applied when the content of the data block or field has a data type "most common value"; "most common value" is determined based on prior knowledge of the data stream. (See id. at 7-8.)

Realtime asserts that all of ISE's accused products use copy encoding, citing Exhibit 2 of the Shamos Declaration. (Opp'n Mot. 4 at 8.) Exhibit 2 does not, however, provide this level of detail. (See Shamos Decl. Ex. 2.) Assuming some of the coding is copy encoding, that is when the content of a data block or field in the current message is analyzed by comparing the content with the content of the corresponding data block or field of a prior message. If two data types are the same, copy encoding is selected as the optimal encoding for the "redundant" data block type of the data block. (Shamos Decl. ¶¶ 17, 71, 73.) When copy encoding is utilized, the question is whether there has been any actual "encoding" at all. According to Dr. Shamos, there has been because a coded representation of the

data block is sent with the message in a PMAP that is included with the message. (See id. ¶¶ 17, 71, 73.) Dr. Shamos also states that some of the accused instrumentalities use increment encoding but he does not explain how. (See id. ¶ 17.)

Finally, according to Realtime, each of ISE's accused instrumentalities use "stop bit encoding" (also called "transfer encoding" or "variable byte encoding"). (Opp'n Mot. 4 at 9.) Stop bit encoding is used by ISE's accused instrumentalities when the data is not suitable for copy, default, or increment encoding. (Shamos Decl. ¶¶ 30, 35.) According to Dr. Shamos, this renders stop bit encoding a content independent compression technique. (See id. ¶ 35.) In addition, Dr. Shamos supports the proposition that "with stop bit encoding, the data field (or block) is converted into a coded representation comprising only the information content bits of the field (or block), which is transmitted along with one or more stop bits that indicate the unused bits of the data field that would not be transmitted." (Opp'n Mot. 4 at 9-10 (citing Shamos Decl. ¶¶ 19, 30, 35).) Thus, according to Dr. Shamos, the "stop bits" along with the information content bits of the encoded data together comprise a coded representation of the data block. (See Shamos Decl. ¶¶ 30, 35.)

ISE argues that stop bit encoding is not, in fact, content independent. (ISE's Reply Br. in Support of its Mot. for Summ.

J. of Noninfringement ("ISE Reply") (Dkt. No. 795) at 7-8.) ISE's System and Products Strategy Officer Maynard states, though, that stop bit encoding embeds intelligence into each byte in a field. (See Maynard Decl. ¶ 20.)

There is a material dispute of fact as to whether the Accused Instrumentalities employ encoders and if so, what type. Accordingly, summary judgment is denied on this basis.

C. Are ISE's Encoders 'Selected' During the Compression Process?

The third basis upon which ISE moves for summary judgment is that, according to ISE, all but two claims (claims 20 and 22 of the '568 Patent) require that an Accused Instrumentality "select" an encoder. (See ISE Mem. at 7.) According to ISE, none of its Accused Instrumentalities meet that requirement.

This Court has previously construed the phrase "selecting an encoder" as "choosing (or choose) an encoder (or lossless encoders) during the compression process based on analyses of content of the data blocks (or data fields)." See Realtime Data, 2012 WL 2394433, at *16. As the Court noted in its Markman opinion, the essential difference between the parties with respect to their proposed constructions has to do with the temporal moment when the encoder is selected. Id. at *12. The Court concluded that the selection occurs during the compression process and that this requires an analysis of the content of the data block. Id. at *13.

According to ISE, the selection of its encoders is preordained--i.e., the encoders are set forth in a template prior to the compression process. Thus, they cannot be selected during the compression process. (ISE Mem. at 8.) According to ISE's Maynard, the selection of the encoder could date back to March 2008 when the template was written. (Maynard Decl. ¶ 12; see also ISE Mem. at 8.)

Realtime disagrees. The essential difference between the parties is whether the encoder "selects" at the time of the creation of the template (which sets forth possible encoders), or whether the selection occurs when the data block is being analyzed during the compression process, when the template is referenced. According to Realtime, ISE's encoders are selected based on an analysis of the content of the data block. Part of that comparison involves an analysis of the data block against the template which sets forth possible encoders for certain data block types. (Opp'n Mot. 4 at 11.)

Realtime argues that "because ISE's section process occurs immediately after analysis, all of ISE's accused products contain the claim limitation." (Opp'n Mot. 4 at 12.) Realtime further argues that the fact that the template offers associations of data block types to optimal encoders does not replace the actual moment of selection of such encoder (even

using the template reference) during the compression process. (Id.)

The debate over the temporal moment of "selection" must be distinguished from other aspects of the selection of the encoder that are not a part of this particular argument for summary judgment. This particular argument is exclusively focused on when selection occurs, not how or whether the descriptor meets required limitations. In terms of this narrow argument, there is a material dispute of fact as to when the selection of the encoder occurs. Therefore, summary judgment is denied on this basis.

D. Does ISE Use a "Descriptor"?

ISE argues that each of the asserted claims, with the exception of claims 20 and 22 of the '568 Patent, requires that the accused instrumentality contain one or more "descriptors," wherein the descriptors indicate which lossless encoder has been selected. (See ISE Mem. at 8.)[9] ISE asserts that none of its Accused Instrumentalities meet this limitation. (Id.)

This Court has previously construed "descriptor with the encoded data which identifies" [the selected encoder], as "recognizable data that is appended to the encoded data for

[9] Claim 1 of the '568 Patent requires, for instance, "providing a descriptor with the encoded data which identifies the selected encoder." ('568 Patent col.23 l.44-45.)

specifying" the selected encoder. See Realtime Data, 2012 WL 2394433, at *16.

The facts relating to this descriptor issue are not in dispute; rather, their characterization is.

ISE argues that its accused instrumentalities do not have a descriptor. ISE contends to get around this issue Realtime tries to construct a "descriptor" through the combination of a PMAP and template ID or a message type. (See ISE Mem. at 9.) ISE argues that either combination does not identify the selected lossless encoders, as required. (Id.)

Dr. Shamos, however, argues that the combination of the PMAP and the template ID provide all of the information that is needed in order to determine which encoder was selected. (See Shamos Decl. ¶ 57.) Dr. Shamos does not, however, state that the information in the template ID or the PMAP in fact identify the selected encoder. Indeed, he could not because they do not.

The PMAP and template ID do not themselves identify the selected encoder; rather, they point to a fixed template which contains various encoder types. The PMAP and template ID then use that template to identify the encoder that would be used optimally to encode the data type at issue. There is no factual dispute that template is not "with" the encoded data; it is referenced by the PMAP and template ID. (See Storer Decl. Mot. 4 ¶ 151.) There is also no real "selection" of an encoder as

part of this process; the encoder is simply referenced from the template.

The critical point of distinction between the parties is whether or not the template can be considered to be "with" or attached to the encoded data. ISE's Maynard argues that it is not--the template is provided to the customer before the compression process even begins and all that is accompanying the data block itself is the template ID and PMAP. (Maynard Decl. ¶¶ 12-17.) Maynard states that the template ID and the PMAP do not themselves provide any indication as to which type of transfer encoding might have been applied to a given piece of data. (Id.)

Realtime argues that ISE has taken a far too narrow view of what a descriptor is--or can be. Realtime does not dispute that the template ID and the PMAP reference back to the template. (Opp'n Mot. 4 at 14.) According to Realtime, the PMAP and template ID are utilized at the decoding end to reference the template which will then indicate which encoder was used. (Id.)

The question on this motion is whether the template, which in fact identifies the encoder that has been used, is "with" the data block as required by the language of the claim. The answer is "no." Although the PMAP and template ID are in fact with the data block, they do not contain information regarding the selected encoder. It takes another step--away from the data

block--to determine which encoder has been selected. The "descriptor" therefore requires both information that is "with" the data block and information that is "not with" the data block. The claim requires that the descriptor be with the data block.

Accordingly, ISE is entitled to summary judgment on this basis.

E. Is ISE's Transfer Encoding Content Independent Data Compression?

ISE argues that claims 22, 25, 26, 29, 34 and 35 of the '651 Patent, claims 14 and 19 of the '747 Patent, and claim 22 of the '568 Patent all require that in order to be infringed, the accused instrumentality must utilize content independent data compression. (ISE Mem. at 10.) According to ISE, none of its Accused Instrumentalities meet this requirement.

This Court has previously construed "content independent data compression" to mean

> compression that is applied to input data that is not compressed with content dependent data compression, the compression applied using one or more encoders without regard to the encoder's (or encoders') ability to effectively encoded the data block type (or data field type).

Realtime Data, 2012 WL 2394433, at *16.

According to ISE, Realtime's infringement contentions (and Dr. Shamos) rely upon stop bit encoding to meet the content independent data compression requirement. (ISE Mem. at 11.)

ISE argues that stop bit encoding--or transfer encoding--is only used on certain data types and therefore must be content dependent. (Id.) According to ISE's Maynard, ISE chooses whether to apply transfer encoding based on the type of data in a field and whether it will be able effectively to encode that data. (Maynard Decl. ¶¶ 27-28.)

Realtime disagrees. According to Realtime, transfer encoding--which ISE concedes it uses--is used when three other types of encoders are not optimal (copy, incremental, and default encoders); that does not mean that the content of the data block is known. In fact, according to Realtime, the content of the data block can nonetheless be one of several different types including fields containing integer numbers, signed integers, unsigned integers, scaled numbers and ASCII strings. (Opp'n Mot. 4 at 16.) Transfer encoding may be applied to other types of data in addition to these but its application may not be optimal. (Id.)

Whether transfer encoding, which is used when other encoding techniques are not optimal, but when the content can be one of a number of different types, is content independent is a question of fact for the jury. Accordingly, the Court denies summary judgment on this basis.

F. Is Stop Bit Encoding "Lossless"?

ISE argues that all of the asserted claims of the '651 and '747 Patents require that the accused instrumentality use a lossless encoder or lossless decoder. (ISE Mem. at 11.)

This Court previously construed "lossless" to mean "technique, software or hardware that fully preserves the original unencoded data such that the decoded data is identical to the unencoded data." Realtime Data, 2012 WL 2394433, at *16.

ISE argues that Realtime has consistently cited ISE's stop bit encoding feature of FAST transfer encoding as meeting the lossless requirement. (ISE Mem. at 11-12.) As a factual matter, ISE argues that Realtime has never shown that any customers who decode ISE's encoded market data get data that is identical to that which was encoded. (Id. at 12.) According to ISE, that amounts to a failure of proof--i.e., that whether the stop bit encoding is lossless or not cannot be demonstrated on the current record and therefore summary judgment is warranted. (Id.)

ISE further argues that even if Realtime had undertaken to show that stop bit encoding was lossless, it could not have. (See ISE Mem. at 12.) ISE's Maynard states that when many feeds are FAST encoded in ISE's system, data is lost. (Maynard Decl. ¶¶ 22-26.) Moreover, when ISE decodes OPRA feeds, that decoding

is not lossless because the encoding was not lossless. (ISE Mem. at 13.)

Realtime first argues that it has not limited itself to ISE's stop bit encoding, and that ISE's accused products, which include but are not limited to stop bit encoding, are lossless. Realtime points to an analysis of source code conducted by Dr. Shamos. (Opp'n Mot. 4 at 17.) According to Realtime, although there may be a lack of total identicality during the transmission process, once the data is encoded losslessly it can be decoded losslessly. (Id.)

Realtime points to ISE's transfer, copy, increment, and default encoding techniques as lossless. Realtime argues that the use of the PMAP and template ID allow for the reconstruction at the decoding end of identical data. Thus, while the data may be transmitted with less than all the data, the decoding process can achieve the required identicality. (Opp'n Mot. 4 at 17.) Realtime is correct that under the Court's construction, identicality is judged as the point of decoding. See Realtime Data, 2012 WL 2394433, at *16.

There is a material issue of fact, however, as to whether ISE's encoding techniques are lossless. Although it may be that there is no record of a before-and-after demonstration of encoding and decoding customer data, given Dr. Shamos' expertise in the field, he can opine as to his views, the jury can credit

his testimony (or not) and weigh that testimony to whatever extent it believes it deserves in light of the analysis Dr. Shamos has (or has) not conducted.

The Court denies summary judgment on this basis.

G. Is there a Data Stream at the Point of Encoding?

ISE argues that all of the asserted claims in the '568 Patent require that the accused encoder must operate on a "data stream." (ISE Mem. at 13.) ISE argues that since none of its Accused Instrumentalities perform this step, none infringe. (Id.)

This Court previously construed "data stream" to mean "one or more blocks transmitted in sequence from an external source whose characteristics are not controlled by the data encoder or decoder." Realtime Data, 2012 WL 2394433, at *16.

ISE's Maynard states that for every ISE Accused Instrumentality, ISE generates its feed internally from stored data. (Maynard Decl. ¶¶ 41-43.) Thus, according to ISE, since the data feeds are "external," they fail to meet one of the construed requirements for a data stream. (ISE Mem. at 13.) ISE points to statements by Dr. Shamos in which he appears to indicate that "ISE feeds" are internal (and therefore not external) to ISE. (Id. at 13-14 (citing Shamos Decl. ¶¶ 45-46).) ISE's Maynard states, however, that ISE's Accused Encoding Instrumentalities encode data that comes from within

ISE's own systems, out of its own trading engines. (Maynard Decl. ¶ 41.) ISE's trading data is always under its own control. (Id.)

According to Realtime, ISE is simply wrong that the trading data is internal. Realtime argues that in fact trading data necessarily derives from external sources: the data from trades in markets or other institutions that are not part of ISE. ISE replies that Dr. Shamos admitted that he had no idea where ISE's trading data came from, and the only evidence in the record is from Maynard who states that it comes entirely from ISE's own internal trading engine.

According to Maynard, ISE's Accused Encoding Instrumentalities encode data that come from within ISE's own systems, and is passed from the trading engine to the server which performs the FAST encoding; that server stores the trading data in memory to create a duplicate set of trading data; the FAST encoder then uses this stored data in the memory of the server to, inter alia, form the FAST feeds. (Maynard Decl. ¶ 41.)

Realtime argues that ISE misconstrues the meaning of "external." (See Opp'n Mot. 4 at 20-21.) Realtime references certain prior art--"Sebastian" (cited in the reexamination proceeding for the '568 Patent)--to show a contrast between encoders that do and do not control change of sequence of data

blocks. (Id. at 21.) In Sebastian, an encoder sorts incoming data blocks into buckets and compresses the data blocks in a particular order; thus, the data stream is not "external" to the encoder. (Id.) Realtime argues that ISE's Accused Instrumentalities do apply the FAST protocol to the data blocks as the data blocks are received and do not control or change the characteristics of the data blocks. (Id.)

The Court agrees that there is no material issue of fact with respect to whether the data is external. The evidentiary record is one sided on this point: the data comes from ISE's own trading engine. Although Realtime has asserted that this must mean that the trades reported in the data originated externally, that is an unsupported factual assertion. At the summary judgment stage, more is required. Thus, there is no material issue of fact that ISE's Encoding Instrumentalities do not infringe claims 15, 20, 22 and 32 of the '568 Patent. Summary judgment is granted on this issue.

H. Does ISE's ISE.FastProcessing Encoding Produce an Output Data Stream?

ISE argues that the encoding claims of both the '651 Patent (claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47 and 49) and the '747 Patent (claims 14, 19) all require that the result or output of encoding be a data stream. (ISE Mem. at 15.) ISE argues that the record is devoid of any evidence that this requirement is met with respect to the ISE.FastProcessing

library. (See id.) According to ISE's Maynard, this library is used exclusively for encoding and decoding journal files, not for creating or decoding data streams. (See Maynard Decl. ¶¶ 35-37.) Journal records are maintained for audit purposes. (Id. ¶ 42.) ISE argues that its Journal Record Application cannot constitute the required "external source"--and that since the journal records are never sent to a third party they are not part of any output data stream. (ISE Mem. at 16.)

Realtime does not dispute that the ISE.FastProcessing Encoding creates journal files. However, according to Realtime, it is of no moment that those journal files are never sent to third parties--there is no requirement that they terminate with a third party. (Opp'n Mot. 4 at 22.) Again, Realtime relies upon its assertion, without record support and as contradicted by Maynard, that the information that forms the trading files must come from some external source. As stated, Maynard is clear that the trading files are generated from ISE's own trading engine and do not come from an external source. (See Maynard Decl. ¶ 41.) Realtime depends upon this assertion to make the following statement:

> Because the activity in its trading engine must be related to information on trades, which are performed externally to ISE's Accused Encoders, ISE's journal records must contain data which was transmitted by an external source. Therefore, ISE's Accused ISE.FastProcessing encoding produces an output 'data stream.'

(Opp'n Mot. 4 at 22.) Realtime does not support its assertion with any citation to any evidence--even an expert opinion. It is mere speculation.

Accordingly, the Court finds that there is no triable issue of fact that the accused ISE.FastProcessing encoding does not infringe the claims recited above. Summary judgment is granted in this regard.

I. Are There Data Packets At the Time of Encoding?

ISE argues that claims 1 (unasserted), 15, and 32 of the '568 Patent require that in order to be infringed, the encoder must operate on data in a "data packet." (ISE Mem. at 16.) ISE claims that it does not in fact encode data packets and therefore cannot infringe claims 1, 15, and 32 of the '568 Patent. (ISE Mem. at 10.)

The parties agree that the term "packet" means "information limited in type, format and content and able to be transmitted as a unit across a packet-switched network, the packet including control information that enables the packet to be delivered to an intended destination in the network." (ISE Mem. at 16.)

ISE asserts that its Encoding Instrumentalities, namely ISE.Fast Encoding, ISE.FastProcessing Encoding and MIDAS Encoding, do not FAST-encode data that is in a data packet. (ISE Mem. at 16.) According to ISE's Maynard, trading data may arrive in packets to the server, but they are stored in memory

and are no longer "in packets" after that. (ISE Reply at 10; Maynard Decl. ¶ 41.) According to ISE, there is no evidence that ISE encodes data in packets at all--and indeed, Maynard states that it does not. (Maynard Decl. ¶ 41.)

Realtime argues that because the trading data is of a specific type (numerical) and format (a format acceptable to the server for which it was intended) and content (records of trades) is transmitted "over some kind of network from the trading engine to a server." (Opp'n Mot. 4 at 23.) Realtime extrapolates from those assertions that this must mean that the data meets the agreed definition of packets and travels in packets in the data stream. This is not supported by any facts. It is unclear that the format is limited in type--Realtime simply asserts without support that it is. In addition, there is nothing supportive of the proposition that the data is being transmitted as a unit, or over a packet-switched network (versus, as Realtime states, some kind of network from the trading engine). (Opp'n Mot. 4 at 23.)

Realtime's arguments do not raise a triable issue as to whether the data is in packets when it enters the data stream. Thus, ISE is entitled to summary judgment as to claims 15 and 32 of the '568 Patent.

J. Does ISE Infringe Decoder Claims If It Does Not Perform the Encoding Step?

ISE asserts that its Accused Decoding Instrumentalities, ISE's Exegy OPRA Decoding, OPRA Decoding and ISE.FastSpec Greeks Decoding cannot infringe claims 95, 97, 108, and 112 of the '651 Patent because they do not perform the required encoding step. (ISE Mem. at 17.)

Each of these decoding claims contains the following limitation: "wherein the lossless encoders are selected based on analyses of content of the data blocks [or fields]." (Id.) The parties have agreed that this phrase as used in these claims means "the system (or method) selects the lossless encoders based on analyses of content of the data blocks (or data fields)." (Id.)

According to ISE, Dr. Shamos concedes that these decoding claims require encoding. (ISE Mem. at 17 (citing Shamos Decl. ¶ 78).) The critical issue is whether to infringe these claims a single party--here, ISE--must itself perform (or directly control another who performs) both the encoding and decoding steps.

ISE argues that there is no evidentiary support--and no proffered opinion by Dr. Shamos--that ISE performs both the encoding and decoding steps for these claims. (ISE Mem. at 18.) Indeed, it cites to Maynard as stating the opposite. (Id.) According to Maynard, OPRA sends a feed to ISE, which OPRA

itself encoded; ISE then uses its OPRA decoding software to decode that feed--but it had nothing to do with the original encoding. (See Maynard Decl. ¶ 44.) There is no argument that ISE controls OPRA.

With respect to the Greeks decoding function, the Volera Greeks feed comes from an unrelated third party company called Hanweek Associates LLC ("Hanweek"). ISE does own 20 percent of Hanweek.

Realtime does not dispute ISE's factual arguments. Instead, it simply asserts that ISE "was the mastermind" behind the performance of all of the steps of these asserted claims. (Opp'n Mot. 4 at 24.) Such an unsupported statement is insufficient at the summary judgment stage.

There is no material issue of fact with respect to whether ISE's OPRA decoder infringes--it does not; it does not perform an essential step of the claim. Accordingly, summary judgment is granted as to these claims.

K. Does ISE's MIDAS Encoding Infringe Claims 20 and 22?

ISE argues that claims 20 and 22 of the '568 Patent require the step of "processing the description file with a data compression compiler, and outputting an executable file that is used to process a stream of data by recognizing data field types in the data stream and applying encoders associated with the

recognized data field types to encode the data stream." (ISE Mem. at 19.)

According to ISE, its accused MIDAS Encoding is based on software obtained from a third party, OMX. (Id. at 19.) ISE's Maynard states that ISE has never possessed the source code for this software and has never compiled it. (See Maynard Decl. ¶¶ 33, 45.) Realtime does not contradict these statements.

Instead, Realtime argues that "ISE contracted with OMX for 'deliver[y] of a production-ready version of the Licensed Product.'" (Opp'n Mot. 4 at 23.) Realtime then relies upon the legal proposition that one cannot avoid infringement by having another perform one of the required steps. (See id. at 24.) That argument is applicable to situations in which another is contractually obligated to perform the claimed steps. See Akamai Techs., Inc. v. Limelight Networks, Inc., 629 F.3d 1311 (Fed. Cir. 2010). Critically, there is more to the claimed step than what can be read as a matter of law or assumed as a matter of fact into the delivery of a production-ready version of the software: the steps require processing the description file with a data compression compiler, and outputting an executable file that is used to process a stream of data by recognizing data field types. (See ISE Reply at 10; '568 Patent col.24 ll.56-67.)

Realtime has not proffered the opinion of an expert that delivery of the software requires compiling; or that any of these other steps are a necessary part of such delivery. Accordingly, there is no triable issue of fact on this issue and ISE is entitled to summary judgment with respect to claims 20 and 22 of the ‘568 Patent.

Conclusion: Motion No. 4

The above rulings complete the Court’s decision on ISE’s motion for summary judgment of noninfringement (Motion No. 4). ISE’s motion for summary judgment is GRANTED IN PART and DENIED IN PART as set forth above.

VI. MOTION 5

Defendants CME Group, Inc., Board of Trade of the City of Chicago, Inc., and the New York Mercantile Exchange, Inc. (collectively the “CME Defendants”) move for summary judgment on the basis that their encoding and decoding systems and methods do not analyze the content of a field or block to determine a data field type or data block type. (Mem. of Law in Support of CME Defs.’ Mot. for Summ. J. of Noninfringement Based on Lack of Determining a Data Block or Field Type (“CME Mem.”) (Dkt. No. 660) at 1.)[10]

[10] The following defendants join in Motion No. 5: The Goldman Sachs Group, Inc., Goldman, Sachs & Co., Goldman Sachs Execution & Clearing, L.P., J.P. Morgan Chase & Co., J.P. Morgan Securities, Inc., J.P. Morgan Clearing Corp., Morgan Stanley, Morgan Stanley & Co, Incorporated, HSBC Bank USA, National Association, HSBC Securities USA, Inc., Credit Suisse Holdings (USA), Inc., Credit Suisse Securities (USA), LLC, Thomson Reuters Corporation, Bloomberg

In its infringement contentions with respect to CME, Realtime asserted that the claim requirement of determining a data type was met because the encoder "checks" a current value of a data field. (CME Mem. at 2.) This Court previously construed data block type and data field type as "categorization of the data in the field (or block) as one of ASCII, image data, multimedia data, signed and unsigned integers, pointers or other data types." Realtime Data, 2012 WL 2394433, at *16.

The crux of the CME Defendants' arguments in support of their motion is that prior to the Markman opinion, and then in connection with its infringement contentions, Realtime has used the concept of "checking a value" rather than categorization of the data. According to the CME Defendants, Realtime has used this language relating to "checking a value" with respect to 7 of the 8 asserted independent claims, and has used similar language for the eighth: claims 13, 22, 29, 60 of the '651 Patent, claims 1 and 20 of the '568 Patent, and claims 14 and 19 of the '747 Patent. (CME Mem. at 6.)[11] They argue that Realtime is wrong in asserting that the requirement of analyzing the "content of the data block to determine data type" is met when the encoder checks for a relation between the current value of

L.P., Interactive Data Corporation, BATS Trading, Inc., BATS Exchange, Inc., The NASDAQ OMX Group, NASDAQ OMX PHLX LLC, Chicago Board Options Exchange, Incorporated, BOX Options Exchange LLC

[11] The law is clear that if independent claims are not infringed, neither are dependent claims. See Minn. Min. & Mfg. Co. v. Chemque, Inc., 303 F.3d 1294, 1300 (Fed. Cir. 2002).

the message field and a value of the corresponding message field in a previous message in the data packet because the Court explicitly reject this construction of the claim in the Markman opinion. (Id. at 6-7.)

Realtime argues that the CME Defendants truncate the steps they are outlining and that they are undertaking a two-step, analytical process: checking a value and then comparing that value. (Realtime Data, LLC's Mem. in Opp'n to CME Defs.' Mot. for Summ. J. of Noninfringement Based on the Lack of Determining a Data Block or Field Type ("Opp'n No. 5") at 9-11.)[12] The CME Defendants respond that this may be the "analysis" step--but it does not answer whether the CME Defendants determine a data block type or data field type during the compression as required by the claim; and the CME Defendants argue that this determining step is more than checking a value. (See Reply in Support of Defs.' Mot. for Summ. J. of Noninfringement Based on Lack of Determining a Data Block or Field Type (11 Civ. 6697, Dkt. No. 800) at 3.)

Realtime grounds its response in the specification language of the patents-in-suit. According to Realtime, a "data type" may be a characteristic of the data block. (Opp'n Mot. 5 at 3.) It is clear from Realtime's description of how it performs this

[12] ISE's motion for summary judgment based on whether an "analysis" occurred was denied, see Part V.4.A. supra; ISE did not move based on content categorization.

step, that it is performing a value check. The question is whether there is an issue of fact as to where this value check can constitute content categorization. There is not. The Court resolved this issue in connection with its Markman opinion.

Realtime argues the CME Defendants' copy encoding technique determines whether the data block is redundant of the corresponding data block in the prior message and that they perform this step through a "content check." (Opp'n Mot. 5 at 4.) This is, as Dr. Storer states (uncontradicted by Dr. Shamos), really a value check--not a content categorization (e.g., determining whether the content is ASCII, multimedia, signed or unsigned integers, etc.).

With respect to its increment encoding technique, Realtime states that the CME Defendants determine the data field (or block) type by analyzing whether the data block has a sequential difference of one. (Id. at 4.) Realtime argues that this sequential difference is also a characteristic of the data block. This is a value check.

Finally, with respect to the CME Defendants' default encoding technique, Realtime asserts that the data field (block) type is also determined by identifying whether the data block has content that is the most common value for that data block or field that is expected for that data block based on a priori knowledge of the data stream. (Id. at 4.) This is also clearly

a species of checking a value and not of content categorization.[13]

The mere repetition of the word "content" throughout Realtime's argument cannot change what the experts opine is occurring: the CME Defendants' encoding techniques check values; they are not examining the content of the data block for its proper categorization.

The CME Defendants are entitled to summary judgment with respect to their Accused Encoding Instrumentalities. Their motion is GRANTED.

VII. MOTION 6

Defendants NYSE Euronext, NYSE ARCA, Inc., NYSE AMEX, LLC and Securities Industry Automation Corporation (collectively "NYSE"), and OPRA (collectively with NYSE, the "NYSE/OPRA Defendants") have moved for summary judgment of non-infringement as to all claims of the patents-in-suit.[14]

According to the NYSE/OPRA Defendants, the encoding and data block type (or field) limitations, which are asserted in every claim they are accused of infringing, and the selecting

[13] As an alternative argument, the CME Defendants assert that they does not infringe because it does not meet the "data block type" or "data field type" limitation of the claims. Here again, the question is whether checking a value is determining a categorization of a data block (or field) type. As stated above, the answer to the question posed in this way (which is only slightly different from that discussed above) is "no." Checking a value is not the same as categorizing data type as, inter alia, ASCII, multimedia, signed and unsigned integers, etc. This is an alternative basis for granting summary judgment.

[14] The same defendants that joined in Motion No. 5 similarly join in Motion No. 6.

limitation, which is asserted in all but two claims asserted against them (claims 20 and 22 of the '568 Patent), are absent from the accused systems. (See NYSE and OPRA Defs.' Mem. of Law in Support of Mot. for Summ. J. of Noninfringement Based on the Absence of Encoding, Data Block (or Field) Type, and Selecting Limitations ("NYSE/OPRA Mem.") at 1.)

The NYSE/OPRA Defendants make two arguments with regard to encoding: (1) that they do not encode since the NYSE and OPRA systems all throw data away and result in no change, which this Court rejected as constituting "encoding" in its Markman opinion; and (2) that the field operators do not generate a coded representation of the incoming data as this Court in its Markman opinion stated was required in order to encode.

As to the data type limitation, the NYSE/OPRA Defendants repeat the arguments made by the CME Defendants in Motion 5 above: that there is no analysis to determine the data type nor is there an analysis of data type in connection with the selection of the encoder. (NYSE/OPRA Mem. at 2.)

In terms of the selecting limitation, the NYSE/OPRA Defendants argue that this is made during the operation of the field operator identified by Realtime as a content dependent encoder. In contrast, the patent covers selection of the encoder itself, which necessarily has to begin before any encoder begins encoding. In addition, according to the

NYSE/OPRA Defendants, the two encoders identified by Realtime are not alternatives--they are in fact applied one after the other. Finally, the NYSE/OPRA Defendants argue that the selection identified in the accused products is based on data value, not data type as the claims require. (Id. at 2.)

A. Analyzing the Content of the Data Block Type

The NYSE/OPRA Defendants make the same arguments as those made by the CME Defendants with respect to the data block (or field) type limitation. They argue that a value check is not the type of data categorization which must occur. Realtime argues that it is, because the "value" is a characteristic of the data, and constitutes "other data type" even under the Court's Markman ruling. Realtime also argues that the NYSE/OPRA Defendants undertake a two step process that does constitute the necessary analysis of the data block (or field) type.

The Court refers to its analysis of these same arguments with respect to Motion 5. The NYSE/OPRA Defendants are correct that there is no triable issue of fact with respect to their data block (or field) type arguments, and they are entitled to summary judgment on this basis as a matter of law.

B. The Encoding Limitation

According to Realtime, the NYSE/OPRA Defendants' field operators do, in fact, convert the contents of a data block (or data field) into a coded representation of those blocks. The

NYSE/OPRA Defendants use copy, default, increment and transfer (or stop bit) encoding techniques. For the same reasons set forth in connection with ISE's motion for summary judgment on this issue (Motion No. 4), there are material issues of fact that preclude summary judgment.

Specifically, according to Realtime's expert, Dr. Shamos, in fact all three of these field operators generate a coded representation. (See Shamos Decl. ¶¶ 17, 40, Ex. 1.) Realtime asserts that these products all use FAST encoding techniques. (Realtime Data, LLC's Mem. in Opp'n to NYSE & OPRA Defs.' Mot. for Summ. J. of Noninfringement Based on the Absence of Encoding, Data Block (or Field) Type, and Selecting Limitations ("Opp'n Mot. 6") at 13.) According to Dr. Shamos, lossless encoding techniques can include "field encoding"--and field encoding includes copy, default, and increment encoding. (Shamos Decl. ¶¶ 17, 30.)

With respect to stop bit encoding, Drs. Shamos and Storer again disagree. According to Dr. Shamos, while data bits may be thrown away in stop bit encoding, and transmitted in that manner, they can be identically replicated at the decoding end. (Shamos Decl. ¶¶ 19, 30, 35.) Dr. Storer opines that what Realtime calls an encoder is not. (Storer Decl. Mot. 6 ¶ 43.)

For the reasons stated in connection with ISE's motion on this same issue, the Court finds there are triable issues of

fact as to whether the encoders in fact "code." Accordingly, the motion is denied on these grounds.

C. The Selecting Limitation

The NYSE/OPRA Defendants claim that the selection limitation cannot be met because it occurs far earlier than "during the compression process" as required. According to Dr. Storer, the selection of an encoder occurs at the time the code for the template is written. (See NYSE/OPRA Mem. at 16 (citing Storer Decl. ¶¶ 21, 23, 29).)

As the Court noted in its Markman opinion, the essential difference between the parties with respect to their proposed constructions has to do with the temporal moment when the encoder is selected. Realtime Data, 2012 WL 2394433, at *12. The Court concluded that the selection occurs during the compression process and that this requires an analysis of the content of the data block. Id. at *13.

ISE made a somewhat similar argument to that being made by the NYSE/OPRA Defendants. In both instances, the question is whether the selection of the encoder occurs during or prior to the compression process. With respect to ISE, the presence of the PMAP and template ID raised a question of fact.

Here, the question of fact is based on Dr. Shamos' contention that the FAST copy encoding technique actually requires directly analyzing the content of each data block (or

data field) to be transmitted; this occurs as the data blocks are present during the compression process, not when the code was written. The Court cannot therefore grant summary judgment on the basis of this argument.

Even if the selection occurs during the compression process, the NYSE/OPRA Defendants argue that the selection limitation is still not met because that selection is never based on data type. (See NYSE/OPRA Mem. at 16-17.) Realtime disagrees. (See Opp'n No. 6 at 20.) Realtime's response on this issue is, however, less than comprehensible. It appears that it is arguing that there is an analysis of data block/field to determine data type and to select an encoder; however, if this is the same type of analysis that is simply checking a value (it is unclear from Realtime's memorandum and supporting declaration), then this is not the type of data block analysis that is required.

The Court assumes that this is in fact what Realtime is arguing and therefore grants summary judgment based on this argument. If the Court has misconstrued Realtime's argument, then it should clarify its position promptly.

VIII. MOTION 8

A group of defendants (the "Motion 8 Defendants")[15] have moved for summary judgment on largely the same basis as set forth in Motion 4(D): that a descriptor is not "with" the data blocks. The Motion 8 Defendants argue that independent claims 14 and 19 of the '747 Patent, claims 13, 22, 29, 43, 91 and 108 of the '651 Patent, and claims 15 and 32 of the '568 Patent (and the claims that depend from these claims) all require descriptors.[16] (See Defs.' Mem. of Law in Support of Motion for Summ. J. of Non-Infringement: Descriptor Limitation ("Mot. 8 Mem.") at 1.)

Here, the Motion 8 Defendants do not argue that the "PMAP plus template ID, referring to an encoder" is not a "descriptor" as ISE does, but rather argue that this combination does not "specify" the encoder as the claims require, at most there is an indirect reference. Those semantic differences aside, the

[15] The Motion 8 Defendants are NYSE Euronext, NYSE ARCA, Inc., NYSE AMEX, LLC, Securities Industry Automation Corporation, OPRA, HSBC Bank USA, N.A., HSBC Securities (USA), Inc., Chicago Board Options Exchange, Incorporated, FactSet Research Systems, Inc, BATS Trading, Inc., BATS Exchange, Inc., Morgan Stanley, Morgan Stanley & Co. Inc., The Goldman Sachs Group, Goldman, Sachs, & Co., Goldman Sachs Execution & Clearing, L.P., J.P. Morgan Chase & Co., J.P. Morgan Securities Inc., J.P. Morgan Clearing Corp., Thomson Reuters Corp., BOX Options Exchange LLC, International Securities Exchange LLC, The NASDAQ OMX Group, Inc., NASDAQ OMX PHLX LLC, Bloomberg, L.P., CME Group Inc., Board of Trade of the City of Chicago, Inc., New York Mercantile Exchange, Inc., Penson Worldwide, Inc., and Nexa Technologies, Inc.

[16] As an example, claim 14 of the '747 Patent requires "providing a descriptor for the compressed data packet in the data stream, wherein the descriptor indicates the one or more selected lossless encoders for the encoded data block."

Motion 8 Defendants' arguments and ISE's (see Part V.D. supra) reduce to the same and thus, are resolved the same way.

This Court construed "descriptor[s] indicate" as the equivalent of "descriptor with . . .": both require that "recognizable data that is appended to the encoded data for specifying". See Realtime Data, 2012 WL 2394433, at *16. The Court specifically analyzed the question of whether the descriptor must be physically attached to the data or can be associated with the data. It determined that it must be attached in the sense of being "with." Thus, the Court rejected the construction that would allow the referenced template, which is associated with the data. The Court explained that its construction was consistent with multiple references in the specification itself which used language such as "with" or "appended". As this Court found, "[T]here is no support in the specification or claims for the descriptor to be completely detached from the data block." Realtime, 2012 WL 2394433, at *15. The requirement that the template ID and PMAP reference a detached template in order to determine which encoder has been used, does not meet the claim limitation.

Accordingly, the Motion 8 Defendants are entitled to summary judgment on this basis.

IX. MOTION 9

Defendants Credit Suisse Holdings (USA), Inc. and Credit Suisse Securities (USA) LLC (collectively, "Credit Suisse") have also moved for summary judgment based on the descriptor limitation.[17] Credit Suisse's description of the process is somewhat different from that described by in Motions 4 and 8, but nonetheless makes the same major points.

In Credit Suisse's motion, it is clear(er) that the template may be sent to the decoding system in advance. In Motions 4 and 8 it was unclear where the template physically resided, but it was clear that it did not reside with the data block which had the template ID and PMAP. Credit Suisse states that "the decoding system receives the Template in advance, and stores a copy of it. Subsequently, upon receipt of an encoded FAST message, this copy of the Template is used to determine which encoders were used to encode the message." (See Credit Suisse's Mem. in Support of Mot. for Summ. J. of Noninfringement: Descriptor Limitation ("CS Mem.") at 3.) As a matter of undisputed fact, the template is not "with" or appended to the data block.

[17] The defendants who joined in the motion: HSBC Bank USA, N.A., HSBC Securities (USA), Inc., Chicago Board Options Exchange, Incorporated, BATS Trading, Inc., BATS Exchange, Inc., Morgan Stanley, Morgan Stanley & Co. Inc., The Goldman Sachs Group, Goldman, Sachs, & Co., Goldman Sachs Execution & Clearing, L.P., J.P. Morgan Chase & Co., J.P. Morgan Securities Inc., J.P. Morgan Clearing Corp., Thomson Reuters Corp., BOX Options Exchange LLC, International Securities Exchange LLC, The NASDAQ OMX Group, Inc., NASDAQ OMX PHLX LLC, Bloomberg, L.P., Interactive Data Corp., CME Group Inc., Board of Trade of the City of Chicago, Inc., and New York Mercantile Exchange, Inc..

It is still a reference tool that is separate and apart from the data block that has been encoded. Accordingly, this process does not meet the required limitations of the claims requiring a descriptor, and summary judgment is therefore granted.

CONCLUSION

The Court's rulings on Motions 4, 5, 6, 8 and 9 are as set forth above.

Motion No. 4 is GRANTED IN PART and DENIED IN PART. Motion No. 5 is GRANTED. Motion No. 6 is GRANTED IN PART and DENIED IN PART. Motion No. 8 is GRANTED. Motion No. 9 is GRANTED.

The parties are directed to submit to the Court a letter within four business days of the issuance of this order setting forth (1) whether and how these rulings affect the upcoming trial in the Exchange Action; and (2) the order that the Court should proceed to resolve the remaining motions for summary judgment.

The Clerk of the Court is directed to terminate the following motions:

11 Civ. 6696: Dkt. No. 523

11 Civ. 6697: Dkt. Nos. 684, 659, 680, 687

11 Civ. 6699: Dkt. Nos. 92, 109, 112

11 Civ. 6701: Dkt. No. 111

11 Civ. 6702: Dkt. Nos. 132, 149, 152

11 Civ. 6704: Dkt. Nos. 114

SO ORDERED:

Dated: New York, New York
September 24, 2012

K. B. Forrest

KATHERINE B. FORREST
United States District Judge

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- MORGAN STANLEY, *et al.*, Defendants.	11 Civ. 6696 (KBF) 11 Civ. 6701 (KBF) 11 Civ. 6704 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- CME GROUP INC., *et al.*, Defendants.	11 Civ. 6697 (KBF) 11 Civ. 6699 (KBF) 11 Civ. 6702 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- THOMSON REUTERS, *et al.*, Defendants.	11 Civ. 6698 (KBF) 11 Civ. 6700 (KBF) 11 Civ. 6703 (KBF) <u>OPINION AND ORDER</u> (Motion 10)

KATHERINE B. FORREST, District Judge:

From June through September 2012, this Court rendered decisions on a number of summary judgment motions made by various defendants in the above-captioned series of sprawling and related patent suits. This Opinion and Order is

the Court's last decision on the numerous and voluminous summary judgment motions at this time.

In keeping with its practices of referring to the summary judgment motions by Court-assigned number, this Court refers to the motion addressed in this Opinion and Order as "Motion No. 10"--the Motion for Summary Judgment of Non-Infringement: Data Stream, Content Independent Encoding and Lossless Claim Limitations by defendants NYSE Euronext, NYSE ARCA, Inc. ("ARCA"), NYSE AMEX, LLC ("AMEX"), Securities Industry Automation Corporation ("SIAC," and with NYSE Euronext, ARCA, and AMEX, "NYSE"), and Options Price and Reporting Authority, LLC ("OPRA").[1]

For the reasons discussed below, the motion is GRANTED IN PART and DENIED IN PART.

I. BACKGROUND

Plaintiff Realtime Data, LLC ("Realtime") has asserted that NYSE and OPRA infringe claims from U.S. Patent No. 7,417,568 (the "'568 Patent"), 7,777,651

[1] Motion No. 10 is joined by the Bank Defendants--The Goldman Sachs Group, Inc.; Goldman, Sachs & Co., Goldman Sachs Execution & Clearing L.P.; J.P. Morgan Chase & Co.; J.P. Morgan Securities, Inc.; J.P. Morgan Clearing Corp.; Morgan Stanley; Morgan Stanley & Co., Incorporated; Credit Suisse Holdings (USA), Inc.; Credit Suisse Securities (USA) LLC; HSBC Bank USA, N.A.; and HSBC Securities (USA), Inc.--as well as defendants Factset Research Systems, Inc., Bloomberg L.P., Interactive Data Corporation, BATS Trading Inc., NASDAQ OMX Group, Inc.; and NASDAQ OMX PHLX, Inc.

Although the parties had requested that the Court rule only upon the "data stream" issue presented in Motion No. 10 as it related to NYSE and OPRA, for sake of completeness, the Court has ruled on all issues presented by the motion--consistent with its rulings in Realtime Data LLC v. Morgan Stanley, et al., --- F. Supp. 2d ---, 2012 WL 4341808 (S.D.N.Y. Sept. 24, 2012).

(the "'651 Patent"), and 7,714,747 (the "'747 Patent," and with the '568 and 651 Patents, the "patents-in-suits").[2]

The Court assumes familiarity with its various prior summary judgment decisions in this matter, see Realtime Data LLC v. Morgan Stanley, et al., --- F. Supp. 2d ---, 2012 WL 4341808 (S.D.N.Y. Sept. 24, 2012) (the "September Opinion" or "Sept. Op."); Realtime Data LLC v. Morgan Stanley et al., 2012 WL 3158196 (S.D.N.Y. Aug. 2, 2012); Realtime Data LLC v. Morgan Stanley, et al., 2012 WL 2545096 (S.D.N.Y. June 27, 2012); Realtime Data LLC v. Morgan Stanley, et al., 2012 WL 2434750 (S.D.N.Y. June 26, 2012), as well as the Court's opinion on claims construction, Realtime Data, LLC v. Morgan Stanley, --- F. Supp. 2d---, 2012 WL 2394433 (S.D.N.Y. June 22, 2012) (the "Markman opinion"). Together, these opinions summarize various aspects of the inventions at issue and provide background helpful to understanding the instant motion.[3]

NYSE and OPRA assert three separate bases for summary judgment, each of which bears similarity to arguments made by different defendants, and addressed in the September Opinion, see Realtime Data LLC, 2012 WL 4341808:

First, NYSE and OPRA urge that all of the asserted claims in the '568 Patent require the analysis of a data field that is in a "data stream" fall short since, prior to compression, each of their Accused Instrumentalities removes data, adds data, or

---

[2] Asserted claims of the '568 Patent against these defendants include 15, 20, 22 and 32. Asserted independent claims of the '651 Patent against these defendants include 13, 22, 29, 43, 91 and 108. Asserted independent claims of the '747 Patent against these defendants include 14 and 19.

[3] Defined terms or proper names used herein have the same meaning as in the Court's September Opinion.

does not transmit the data blocks in sequence. ISE moved for summary judgment on the basis that the required analysis of the data field in the data stream was not met. (See, e.g., Mem. of Law in Support of ISE's Mot. for Summ. J. of Noninfringement (11 Civ. 6697, Dkt. No. 685) ("ISE Mem. Mot. 4") at Arguments A and G.) With respect to ISE, the Court found that on the record before it, summary judgment was warranted based on ISE's Argument G. See Sept. Op., 2012 WL 4341808, at *10-11. Based upon the record relating to NYSE and OPRA, this Court finds that Realtime has not raised a triable issue of fact as to the particular arguments NYSE and OPRA have made in this regard and thus, that summary judgment is warranted.

Second, certain claims of the '747 and '651 Patents require the application of content independent encoding ("CIC"). OPRA and NYSE argue that the transfer encoding they employ is not content independent. This is similar to the argument made and by ISE and determined by the Court in the portion of the September Opinion relating to Motion No. 4. Sept. Op., 2012 WL 4341808, at *5-6. The Court denied summary judgment to ISE based on a similar argument as that pursued by NYSE and OPRA here.

According to NYSE and OPRA, their Accused Instrumentalities do not utilize CIC. However, there are expert declarations submitted on both sides of this issue-- and the Court cannot resolve the disputed issues on a motion for summary judgment. Accordingly, summary judgment on this basis is denied.

Third and finally, NYSE and OPRA argue that certain claims of both the '747 and '651 Patents require "lossless" encoding or decoding--and that their Accused Instrumentalities use "stop bit encoding" which is not lossless. This is similar to an argument raised by ISE in Argument F in Motion No. 4. There, the Court found that expert testimony on this issue raised a material issue of fact properly decided by a jury. Sept. Op., 2012 WL 4341808, at *9-10. Here, similar issues preclude summary judgment.

II. DISCUSSION

A. Legal Standard

Summary judgment is warranted if the pleadings, the discovery and disclosure materials, along with any (admissible) affidavits, demonstrate that there is no genuine issue of fact necessitating resolution at trial. Fed. R. Civ. P. 56(c); see also Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 247 (1986); Celotex Corp. v. Catrett, 477 U.S. 317, 322-323 (1986). A party moving for summary judgment bears the initial burden of demonstrating that no genuine issue of material fact exists; all reasonable inferences should be drawn in favor of the non-moving party. See Liberty Lobby, 477 U.S. at 255; Cont'l Can Co. USA, Inc. v. Monsanto Co., 948 F.2d 1264, 1265 (Fed. Cir. 1991). The burden then shifts to the non-moving party to come forward with "admissible evidence sufficient to raise a genuine issue of fact for trial in order to avoid summary judgment." Jaramillo v. Weyerhauser Co., 536 F.3d 140, 145 (2d Cir. 2008); see also Glaverbel Societe Anonyme v. Northlake Mktg. & Supply, Inc., 45 F.3d 1550, 1560-61 (Fed. Cir. 1995) ("When the movant's burden of

establishing the lack of a genuine issue of material fact has been met 'in facial terms,' the nonmovant must point to 'some evidence in the record sufficient to suggest that his view of the issue might be adopted by a reasonable factfinder.'" (quoting Resolution Trust Corp. v. Juergens, 965 F.2d 149, 151 (7th Cir. 1992))). Where the non-moving party would bear the ultimate burden of proof on an issue at trial, the moving party satisfies its burden on the motion by pointing to an absence of evidence to support an essential element of the non-movant's claim. See Intellicall, Inc. v. Phonometrics, Inc., 952 F.2d 1384, 1389 (Fed. Cir. 1992).

Where it is clear that no rational trier of fact could find in favor of the non-moving party, summary judgment is warranted. See Liberty Lobby, 477 U.S. at 248. However, the mere possibility that a dispute may exist, without more, is not sufficient to overcome a convincing presentation by the moving party. Id. at 247-48. Similarly, mere speculation or conjecture is insufficient to defeat a motion. W. World Ins. Co. v. Stack Oil, Inc., 922 F.2d 118, 121 (2d Cir. 1990).

In ruling on a motion for summary judgment, a court cannot, however, weigh the evidence or make credibility determinations: those are the functions of the jury. See Liberty Lobby, 477 U.S. at 255. Further, when there are dueling experts, both of whom have put forward opinions in contradiction with each other, and when those opinions are important to resolution of a material factual dispute, summary judgment may not be appropriate. See Hodosh v. Block Drug Co., Inc., 786 F.2d 1136, 1143 (Fed. Cir. 1986) ("The fact issues herein must be resolved by trial in

which the conflicting views of the experts will be subject to the refining fire of cross examination.").

The question is whether, at this stage of the proceeding, the court can determine whether the expert is merely asserting his own ipse dixit, which would be insufficient to defeat summary judgment, or whether two experts in the field could have reasonable differences. If it is the latter, then the Court must leave credibility determinations and the weighing of the experts' opinions to the jury.

B. Do the Accused Instrumentalities Meet the "Data Stream" Requirement?

In the Markman opinion, the Court construed the term "data stream" as requiring "one or more blocks transmitted in sequence from an external source whose characteristics are not controlled by the encoder or decoder." See Markman opinion, 2012 WL 2394433, at *16.

There are four different encoders that Realtime asserts infringe claims 15, 20, 22 and 32 of the '568 Patent: OPRA Encoding, ArcaBook Encoding, Filtered Options Feed Encoding, and XDP Depth of Book.[4] Each of those Accused Instrumentalities has its own technology that must be considered in connection with an infringement claim:

1. OPRA Encoding

(a) Internal OPRA encoding: the OPRA system receives a message from PartiApps; PartiApps deblocks the message, converts it into binary format, inserts a time

[4] Realtime has not pursued its claim with respect to XDP Depth of Book; the Court therefore will not address that product further.

stamp message, and regroups the messages by option symbol. After these steps have all occurred, the message now has internally sourced data and sequencing changes. (See Decl. of Robert Jakob in Supp. of NYSE and OPRA's Mot. for Summ. J. (Dkt. No. 702) ("Jakob Dec.") ¶¶14-15, Ex. A.)

(b) High Speed Line Applications ("HSlApps") receive the message from PartiApps. Prior to those messages undergoing FAST compression, they are altered again: HS1Apps are deblocked, a time stamp is inserted, a sequence number is inserted, and a retransmission requester field is inserted. The messages are then regrouped. As with PartiApps, the HSlApps now have internally sourced data and sequence changes.

2. ArcaBook Encoding

(a) ArcaBook for Equities utilizes both front- and back-end servers. When a message is received at its front-end server, the server adds a sequence number to each message and places the message into the send buffer. As a result, the characteristics of the data have been internally altered; when these messages get to the back-

end server they are therefore different from what was externally received by the front-end server.

(b) ArcaBook for Options works differently--it does not have a front-end server. The messages that ArcaBook for Options compresses are generated by the back-end server itself; the messages therefore do not come from an external source.

3. Filtered Options Feed

(a) OPRA's Filtered Option Feed receives data encoded in the OPRA FAST format. Prior to compression, the feed decodes the OPRA FAST message, creates message objects (including by discarding certain objects), and inserts sequence numbers into the message. As a result, the messages that enter the Filtered Option Feed from the external source are internally altered prior to compression

NYSE and OPRA correctly point out that Realtime fails to make any specific evidentiary showing with respect to whether the three specific Accused Instrumentalities encode data that is in the form of a data stream at the time of encoding: this includes, for instance, an absence of any specific facts relating to how any of the specific encoding products work.

Realtime relies on the generalized statements of its expert, Dr. Ian Shamos. In paragraphs 20-22 of that declaration, Dr. Shamos provided general opinions regarding the data stream requirement for all Accused Products (from any of the dozens of defendants). There is simply nothing in Dr. Shamos' declaration that specifically addresses the NYSE and OPRA data streams, the content of those streams, or the particular way in which their specific encoders act on the data.

The descriptions of NYSE and OPRA's three Accused Instrumentalities here are therefore supported by what is basically an unrebutted factual declaration by defendants' expert: The descriptions of the Accused Instrumentalities all indicate that the data that enters those instrumentalities is internally altered in some fashion and out of the original sequence. That fact is therefore taken as true; there is no triable issue of fact. These three Accused Instrumentalities cannot meet the definition of "data stream" necessary for the asserted claims in the '568 Patent. Summary judgment is granted on this basis.

C. Is Transfer Encoding Content Independent?

This Court has previously construed "content independent data compression" to mean

> compression that is applied to input data that is not compressed with content dependent data compression, the compression applied using one or more encoders without regard to the encoder's (or encoders') ability to effectively encoded the data block type (or data field type).

Markman opinion, 2012 WL 2394433, at *16.

According to NYSE and OPRA (and as ISE argued in Motion No. 4), Realtime's infringement contentions (and Dr. Shamos) rely upon stop bit encoding

to meet the content independent data compression requirement. Defendants argue that stop bit encoding--or transfer encoding--is only used on certain data types and not others and therefore must be content dependent. According to Dr. Storer, on whose declaration defendants rely, stop bit encoding always results in an expansion of the data block. Dr. Storer refers to the "principal creator of FAST, Rolf Anderson" for the point that "byte vectors" are used when stop bit encoding is not optimal. (Decl. of James Storer in Support of NYSE & OPRA's Mot. for Summ. J. ("Storer Decl. Mot. 10") Ex. C at 7.) Moreover, section 10.6 of the FAST standard also states that stop bit encoding is useful for only certain data types. (Id. ¶¶18, 21-22, Ex. B. §10.6.)

Realtime responds to defendants' arguments in a manner similar to its response to ISE's motion: transfer encoding--which defendants concede they use--is used when other types of encoders are not optimal; that does not mean that the content of the data block is known. In fact, according to Realtime, the content of the data block can nonetheless be one of several different types including fields containing integer numbers, signed integers, unsigned integers, scaled numbers and ASCII strings. The debate between the parties can be characterized as whether transfer or stop bit encoding must work with all types of content in order for it to be considered "content independent." This is not a question this Court can resolve on the record before it.

Whether transfer encoding--which is used when other encoding techniques are not optimal, but when the content can be one of a number of different types--is

content independent is a question of fact for the jury. Accordingly, the Court denies summary judgment on this basis.

D. Is Stop Bit Encoding Lossless?

Defendants argue that each of the asserted claims in the '651 and '747 Patents require lossless encoding or decoding. This Court previously construed "lossless" to mean "technique, software or hardware that fully preserves the original unencoded data such that the decoded data is identical to the unencoded data." Realtime Data, 2012 WL 2394433, at *16. The heart of the dispute on this issue is not whether stop bit encoding adds or drops bits--the issue is whether once stop bit encoding has been applied, would the decoded data be identical to the original encoded data.

Defendants argue, without citation to an expert declaration or any other evidence, that "it is impossible to losslessly decode data that was not losslessly encoded." That may or may not be true--the Court certainly cannot make that judgment at this stage of the proceeding. In contrast, Dr. Shamos' declaration in support of Realtime's position (and motion for summary judgment on infringement) does claim that data that has been stop bit encoded can be decoded with the result being bit for bit identical data. (Shamos Decl. ¶ 35.)

Accordingly, there is a triable issue of fact as to whether stop bit encoding is lossless and summary judgment is denied on this basis.

III. CONCLUSION

For the reasons set forth above, summary judgment is GRANTED to NYSE and OPRA as to all asserted claims of the '568 Patent; summary judgment is DENIED on the bases set forth above relating to the '651 and '747 Patents.

The Clerk of the Court is directed to terminate the motion at

- 11 Civ. 6697, Dkt. No. 697
- 11 Civ. 6699, Dkt. No. 123
- 11 Civ. 6702, Dkt. No. 165

SO ORDERED:

Dated: New York, New York
November 9, 2012

K. B. Forrest

KATHERINE B. FORREST
United States District Judge

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, Plaintiff, v. CME GROUP INC., et al., Defendants.	§	No. 11 Civ. 6697 (KBF) No. 11 Civ. 6699 (KBF) No. 11 Civ. 6702 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, v. THOMSON REUTERS CORPORATION, et al., Defendants.	§	No. 11 Civ. 6698 (KBF) No. 11 Civ. 6700 (KBF) No. 11 Civ. 6703 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, v. MORGAN STANLEY, et al., Defendants.	§	No. 11 Civ. 6696 (KBF) No. 11 Civ. 6701 (KBF) No. 11 Civ. 6704 (KBF)

SUPPLEMENTAL ORDER

The Court issued an Opinion & Order in the above-captioned actions (the "Actions") on September 24, 2012 (D.I. No. 843 in Case No. 11 Civ. 6697; D.I. No. 456 in Case No. 11 Civ. 6698; D.I. No. 575 in Case No. 11 Civ. 6696) (the "September 24, 2012 Order") relating to

Summary Judgment Motions Nos. 4, 5, 6, 8 and 9,[1] finding that all of the accused instrumentalities fail to meet one or more of the elements of each asserted claim of the patents-in-suit – U.S. Patent Nos. 7,417,568 (the "'568 Patent"); 7,714,747 (the "'747 Patent"); and 7,777,651 (the "'651 Patent").

The Court issued Orders on September 25, 2012 and October 3, 2012 (D.I. Nos. 844 and 851 in Case No. 11 Civ. 6697; D.I. Nos. 457 and 459 in Case No. 11 Civ. 6698; D.I. Nos. 576 and 578 in Case No. 11 Civ. 6696) clarifying that the September 24, 2012 Order should reflect that Defendant FactSet Research Systems Inc. ("FactSet") joined in Motion Nos. 4 (as it relates to claims 95, 97, 108 and 112 of the '651 Patent), 5, 6, and 8; and that Defendant International Securities Exchange ("ISE") joined in Motion No. 5.

IT IS HEREBY ORDERED that summary judgment is granted in part with respect to the Motion for Summary Judgment of Non-Infringement: Data Stream, Content Independent Encoding and Lossless Claim Limitations filed by Defendants NYSE Euronext, NYSE Arca, Inc., NYSE Amex, LLC, and Securities Industry Automation Corporation (collectively, "NYSE") and Options Price Reporting Authority ("OPRA") (Motion No. 10), as set forth in its Opinion and Order, dated November 8, 2012. Therefore, NYSE's and OPRA's accused encoding instrumentalities do not meet the "data stream" claim element of claims 15, 20, 22 and 32 of the '568 Patent, because those accused encoding instrumentalities encode data that comes from within NYSE's and OPRA's own systems. (*See* September 24, 2012 Order at 25-27.)

IT IS FURTHER ORDERED that summary judgment is granted in part with respect to Defendants NYSE's and OPRA's Motion for Summary Judgment of Non-Infringement for Failure of Proof (Joint Infringement, Acts of Infringement in the U.S. During the Patent Term,

[1] "Motion No." refers to the Motion for Summary Judgment number assigned by the Court in its August 15, 2012 email to the parties, a copy of which is attached hereto as Exhibit 1.

and Mootness) (Motion No. 3), because the Court has found that to infringe claims 95, 97, 108 and 112 of the '651 Patent "a single party . . . must itself perform (or directly control another who performs) both the encoding and decoding steps" (September 24, 2012 Order at 31-32), and for NYSE's accused decoding instrumentalities at issue in that motion, Realtime does not allege that NYSE performs (or directly controls another who performs) the limitation, "wherein the lossless encoders are selected based on analyses of content of the data blocks," of claims 95, 97, 108 and 112 of the '651 Patent.

IT IS FURTHER ORDERED that summary judgment is granted with respect to the Downstream Defendants' Motion for Summary Judgment of Non-Infringement of the Asserted Encoding-Decoding Claims On The Grounds That Downstream Defendants Do Not Select Encoders (Motion No. 1) (D.I. No. 520 in Case No. 11 Civ. 6696) filed by Defendants Morgan Stanley and Morgan Stanley & Co., Inc. (collectively, "Morgan Stanley"); The Goldman Sachs Group, Inc., Goldman Sachs & Co. and Goldman Sachs Execution & Clearing, L.P. (collectively, "Goldman Sachs"); J.P. Morgan Chase & Co., J.P. Morgan Securities Inc. and J.P. Morgan Clearing Corp. f/k/a Bear, Stearns Securities Corp. (collectively, "J.P. Morgan"); and Thomson Reuters Corporation ("Thomson Reuters"), and joined by Defendants BATS Trading, Inc. and BATS Exchange, Inc. (collectively, "BATS") (D.I. No. 708 in Case No. 11 Civ. 6697); NASDAQ OMX Group, Inc. and NASDAQ OMX PHLX, Inc.[2] (collectively, "NASDAQ") (D.I. No. 713 in Case No. 11 Civ. 6697); HSBC Bank USA, N.A. and HSBC Securities (USA), Inc. (collectively, "HSBC"); FactSet; Bloomberg L.P. ("Bloomberg"); and Interactive Data Corporation ("IDC"), as well as Credit Suisse's Motion For Summary Judgment Of Noninfringement On The Ground That Credit Suisse Does Not Select Encoders (D.I. No. 530 in

[2] NASDAQ OMX PHLX, Inc. is now known as NASDAQ OMX PHLX LLC.

Case No. 11 Civ. 6696) (Motion No. 2) filed by Defendants Credit Suisse Holdings (USA), Inc. and Credit Suisse Securities (USA) LLC (collectively, "Credit Suisse"), because the Court has found that to infringe claims 95, 97, 108 and 112 of the '651 Patent "a single party . . . must itself perform (or directly control another who performs) both the encoding and decoding steps" (September 24, 2012 Order at 31-32), and for the Defendants' accused decoding instrumentalities at issue in those motions and joinders, Realtime does not allege that the respective Defendant performs (or directly controls another who performs) the limitation, "wherein the lossless encoders are selected based on analyses of content of the data blocks," of claims 95, 97, 108 and 112 of the '651 Patent.

IT IS FURTHER ORDERED that summary judgment is granted to ISE for non-infringement of claims 15 and 32 of the '568 Patent, claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47, 49, 95, 97, 108 and 112 of the '651 Patent, and claims 14 and 19 of the '747 Patent for the same reasons that the Court granted summary judgment with respect to Motion No. 6 (*see* September 24, 2012 Order at 42-43), because none of ISE's accused instrumentalities meets the claim element "selecting an encoder," "the lossless encoders are selected," "selecting one or more lossless encoders," or "select one or more lossless encoders."

The Clerk of the Court is directed to terminate the following motions:

- 11 Civ. 6696: Dkt. Nos. 520, 530
- 11 Civ. 6697: Dkt. No. 677
- 11 Civ. 6699: Dkt. No. 105
- 11 Civ. 6701: Dkt. Nos. 108, 116
- 11 Civ. 6702: Dkt. No. 145
- 11 Civ. 6704: Dkt. Nos. 111, 119

SO ORDERED:

Dated: New York, New York
November 9, 2012

K. B. Forrest

KATHERINE B. FORREST
UNITED STATES DISTRICT JUDGE

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- MORGAN STANLEY, et al., Defendants.	11 Civ. 6696 (KBF) 11 Civ. 6701 (KBF) 11 Civ. 6704 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- CME GROUP INC., et al. Defendants.	11 Civ. 6697 (KBF) 11 Civ. 6699 (KBF) 11 Civ. 6702 (KBF)
REALTIME DATA, LLC d/b/a IXO, Plaintiff, -v- THOMSON REUTERS, et al. Defendants.	11 Civ. 6698 (KBF) 11 Civ. 6700 (KBF) 11 Civ. 6703 (KBF) AMENDED OPINION & ORDER

KATHERINE B. FORREST, District Judge:

In 2009, plaintiff Realtime Data, LLC d/b/a IXO ("Realtime") sued a number of companies involved in some aspect of the financial services industry (including banks, exchanges, and information services) for infringing three of its patents: U.S. Patent No. 7,417,568 (the "'568 Patent"), U.S. Patent No. 7,714,747 (the "'747 Patent"), and U.S. Patent No. 7,777,651

(the "'651 Patent" and collectively with the '568 and '747 Patents, the "patents-in-suit").

This Court has already issued several decisions on various issues in this litigation, and refers to those decisions for additional facts. See, e.g., Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 3158196 (S.D.N.Y. Aug. 2, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 2545096 (S.D.N.Y. June 27, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 2434750 (S.D.N.Y. June 26, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 2394433 (S.D.N.Y. June 22, 2012); Realtime Data, LLC v. Morgan Stanley, 11 Civ. 6696, 2012 WL 1711117 (S.D.N.Y. May 10, 2012).

Defendants have counterclaimed for non-infringement and invalidity. On July 19, 2012, the Court ordered that the trial as to the Exchange Defendants will proceed first.[1] (See 11 Civ.

---

[1] The "Exchange Defendants" are NYSE Euronext, NYSE ARCA, Inc., NYSE AMEX, LLC, Securities Industry Automation Corporation, Options Price Reporting Authority, LLC, International Securities Exchange, Boston Options Exchange Group LLC, CME Group Inc., Board of Trade of the City of Chicago, Inc., New York Mercantile Exchange, Inc., Chicago Board Options Exchange, Incorporated, BATS Trading, Inc., NASDAQ OMX Group, Inc., and NASDAQ OMX PHLX, Inc.

The term "Bank Defendants" refers to BNY ConvergEx Group LLC, BNY ConvergEx Execution Solutions LLC, Credit Suisse Holdings (USA), Inc., Credit Suisse Securities (USA) LLC, The Goldman Sachs Group, Inc., Goldman Sachs & Co., Goldman Sachs Execution & Clearing, L.P., HSBC Bank USA, N.A., HSBC Securities (USA), Inc., J.P. Morgan Chase & Co., J.P. Morgan Securities, Inc., J.P. Morgan Clearing Corp., Morgan Stanley, and Morgan Stanley & Co., Inc.

The term "Data Provider Defendants" refers to Thomson Reuters Corporation, Factset Research Systems, Inc., Bloomberg L.P., Interactive Data Corporation, Penson Worldwide, Inc., and Nexa Technologies, Inc.

6696, Dkt. No. 539.) That jury trial is scheduled to commence on November 26, 2012.

I. PROCEDURAL BACKGROUND

Defendants and plaintiff now have 19 separate, fully briefed motions for summary judgment (on the question of liability) pending before this Court.[2] The motions put forth a smorgasbord of separate arguments. Each motion asserts more than one basis for the proposition that as to one or more of the accused instrumentalities, Realtime cannot prove a required claim limitation. Accordingly, the 19 motions require the Court to evaluate the merits of far more than 19 arguments.[3]

Buried in one of those motions are issues amenable to resolution on summary judgment; however, many are not. It is important that a jury not be burdened with arguments that are properly resolved by the Court now. It is even more important that this Court not usurp the role of the jury by making determinations of fact, weighing evidence, or making credibility determinations.

[2] Although each of the motions was brought by a particular defendant (or group of defendants), a number of defendants have "joined" the motions of others. The Court notes such joinder in connection with the discussion of each motion addressed in this Opinion.

[3] Given the number of motions pending before the Court in these actions and given the pre-trial and trial schedule in the Exchange Action, for the sake of efficiency and expediency the Court has occasionally cited to the parties' briefs for substantive points (which reference admissible evidence) rather than to the voluminous evidence (that is has reviewed and considered in connection with the pending motions).

Finding a path evenly distributed between judicial efficiency and the merits has been challenging. To assist the Court in sheer organization of the volume of paper, this Court assigned a number to each motion, and requested that the parties submit their views as to the order in which the Court should resolve the motions.

Based upon the Court's review of the letters submitted pursuant to that request (see 11 Civ. 6697, Dkt. Nos. 834 & 835), as well as its own review of the motions and the likelihood that resolving certain motions might assist in more rational trial preparation, this Court resolves the following motions in this Opinion: International Securities Exchange's ("ISE's") Motion for Summary Judgment of Noninfringement (Motion No. 4); CME Group Inc.'s ("CME") Motion for Summary Judgment of Noninfringement Based on the Lack of Determining a Data Block of Field Type (Motion No. 5); NYSE and Options Price Reporting Authority, LLC ("OPRA")'s Motion for Summary Judgment of Noninfringement Due to the Absence of the Encoding, Data Block (or Field) Type, and Selecting Limitations (Motion No. 6); NYSE, et al.'s Motion for Summary Judgment of Non-Infringement Descriptor Limitation (Motion No. 8); and Credit Suisse's Motion for Summary Judgment of Noninfringement: Descriptor Limitation

(Motion No. 9).[4] For ease of reference, the Court refers to the various motions by assigned number throughout this Opinion. When referring to all five motions collectively, the Court uses the term "Motions."[5]

II. THE PATENTS-IN-SUIT

Each Motion delves deeply into claim limitations of the patents-in-suit. Resolution of the Motions requires some understanding of the inventions at issue read against the backdrop of this Court's prior ruling on claim construction. See generally Realtime Data, LLC, 2012 WL 2394433 (referred to herein as the "Markman opinion").

Both the '651 and '568 Patents are entitled "System and Method for Data Feed Acceleration and Encryption." (See '568 Patent at [54]; '651 Patent at [54].) The '568 Patent issued first and discloses a system and methods for providing accelerated transmission of broadcast data, such as financial data and news feeds, over a communication channel using data compression and decompression to increase bandwidth and reduce latency. (See '568 Patent col.5 ll.25-32) The claims of the '568 Patent relate generally to methods for compressing data

[4] Resolution of the other pending motions for summary judgment will be resolved in subsequent opinions.

[5] The Court follows the order in which the parties set forth the arguments in their briefs, as well as the lettering system used by the parties for each of their disparate arguments in the respective Motions.

through encoding applied to a data stream. Certain claims relate to the method and application of selected encoders.

In contrast, the '651 Patent relates to a method of decoding one or more encoded messages. The decoding method requires receiving an encoded message, determining which decoder to utilize, and performing the decoding.

The '747 Patent claims both methods for decompressing one or more compressed data packets using multiple decoders that apply lossless decompression techniques, as well as methods for using multiple encoders that apply lossless compression techniques.

Key aspects of the inventions relate to, inter alia, how and when encoders and decoders are selected, how and when they are applied, to what they are applied, and whether they are in fact lossless.

Defendants universally claim that they have adopted an industry protocol referred to a compression and decompression standard called "FAST," and that FAST does not infringe on plaintiff's patents. Further, the "FIX Protocol" was copyrighted in 2006. It has been utilized by defendants.

III. GENERAL BACKGROUND

In connection with the Motions, there are certain facts relating to the FAST standard that are not materially in dispute.

A basic and overriding contention in this litigation is that FAST infringes the patents-in-suit. FAST is described in the Abstract of the FIX Protocol as "a space and processing efficient encoding method for message oriented data streams. It defines the layout of a binary representation and the semantics of a control structure called a template . . . ." (See Decl. of James Storer In Support of NYSE and OPRA Defs.' Mot. for Summ. J. of Noninfringement Based on the Absence of Encoding, Data Block (Or Field Type), and Selecting ("Storer Decl. Mot. 6") Ex. 1 at NYSE00083123.) According to the FIX Protocol, the FAST encoding method

> reduces the size of a data stream on two levels. First, a concept referred to as Field Operators allows the data affinities of a stream to be leveraged and redundant data to be removed. Second, serialization of the remaining data is accomplished through binary encoding which draws on self-describing field lengths and bit maps indicating the presence or absence of fields.

(Id. at NYSE00083128.)

FAST uses a template, or table, that identifies for each field the data type expected to be present in that field, and a field operator[6] that will be applied to the field during the encoding process. (See Decl. of James Storer in Support of Defs.' Mot. for Summ. J. of Non-Infringement: Descriptor Limitation ("Storer Decl. Mot. 8.") ¶ 11.) The template is not

[6] Realtime calls a "field operator" an encoder; defendants disagree that a field operator is an encoder. The Court need not resolve that issue in order to render its decision on the Motions.

attached to the data block that is compressed. Instead, it acts as a reference tool to identify the appropriate field operator. (See id. ¶ 19.) A template identifier (or "template ID") along with a presence map (or "PMAP") are used to reference the template and determine which field operator was used. (Id. ¶¶ 21-23.) The template ID does not itself state which field operator was used. (Id. ¶ 23.) The template is never transmitted with the compressed data from encoder to decoder. (Id. ¶ 21.)

IV. LEGAL STANDARD

Summary judgment is warranted if the pleadings, the discovery and disclosure materials, along with any (admissible) affidavits, demonstrate that there is no genuine issue of fact necessitating resolution at trial. Fed. R. Civ. P. 56(c); see also Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 247 (1986); Celotex Corp. v. Catrett, 477 U.S. 317, 322-323 (1986). A party moving for summary judgment bears the initial burden of demonstrating that no genuine issue of material fact exists; all reasonable inferences should be drawn in favor of the non-moving party. See Liberty Lobby, 477 U.S. at 255; Cont'l Can Co. USA, Inc. v. Monsanto Co., 948 F.2d 1264, 1265 (Fed. Cir. 1991). The burden then shifts to the non-moving party to come forward with "admissible evidence sufficient to raise a genuine issue of fact for trial in order to avoid summary judgment." Jaramillo v.

Weyerhauser Co., 536 F.3d 140, 145 (2d Cir. 2008); see also Glaverbel Societe Anonyme v. Northlake Mktg. & Supply, Inc., 45 F.3d 1550, 1560-61 (Fed. Cir. 1995) ("When the movant's burden of establishing the lack of a genuine issue of material fact has been met 'in facial terms,' the nonmovant must point to 'some evidence in the record sufficient to suggest that his view of the issue might be adopted by a reasonable factfinder.'" (quoting Resolution Trust Corp. v. Juergens, 965 F.2d 149, 151 (7th Cir. 1992))). Where the non-moving party would bear the ultimate burden of proof on an issue at trial, the moving party satisfies its burden on the motion by pointing to an absence of evidence to support an essential element of the non-movant's claim. See Intellicall, Inc. v. Phonometrics, Inc., 952 F.2d 1384, 1389 (Fed. Cir. 1992).

Where it is clear that no rational trier of fact could find in favor of the non-moving party, summary judgment is warranted. See Liberty Lobby, 477 U.S. at 248. However, the mere possibility that a dispute may exist, without more, is not sufficient to overcome a convincing presentation by the moving party. Id. at 247-48. Similarly, mere speculation or conjecture is insufficient to defeat a motion. W. World Ins. Co. v. Stack Oil, Inc., 922 F.2d 118, 121 (2d Cir. 1990).

In ruling on a motion for summary judgment, a court cannot, however, weigh the evidence or make credibility determinations:

those are the functions of the jury. See Liberty Lobby, 477 U.S. at 255. Further, when there are dueling experts, both of whom have put forward opinions in contradiction with each other, and when those opinions are important to resolution of a material factual dispute, summary judgment may not be appropriate. See Hodosh v. Block Drug Co., Inc., 786 F.2d 1136, 1143 (Fed. Cir. 1986) ("The fact issues herein must be resolved by trial in which the conflicting views of the experts will be subject to the refining fire of cross examination.").

The question is whether, at this stage of the proceeding, the court can determine whether the expert is merely asserting his own ipse dixit, which would be insufficient to defeat summary judgment, or whether two experts in the field could have reasonable differences. If it is the latter, then the Court must leave credibility determinations and the weighing of the experts' opinions to the jury.

V. MOTION 4[7]

ISE is accused of infringing both the '568 and '651 Patents, and has moved for summary judgment as to all claims of infringement asserted against it. It presents 11 arguments supported by an expert declaration from Dr. James A. Storer ("Storer Decl. Mot. 4") and a declaration from ISE's System and Product Strategy Officer, Gregory J. Maynard ("Maynard Decl.").

[7] Defendant FactSet, Inc. joins in Motion No. 4 as it relates to claims 95, 97, 108, and 112 of the '651 Patent.

In opposition, Realtime offers multiple responses to each of the 11 arguments as well as what it believes is support for its position(s) from the declaration of its own expert, Dr. Michael Ian Shamos ("Shamos Decl.").

According to Realtime, three of ISE's encoding products and methods are infringing instrumentalities: (1) ISE.Fast Encoding; (2) MIDAS Encoding; and (3) ISE.FastProcessing Encoding (collectively, the "ISE Accused Encoding Instrumentalities"). Realtime also accuses four of ISE's decoding products or methods of being infringing instrumentalities: (1) ISE.FastProcessing decoding; (2) Exergy OPRA Decoding; (3) OPRA Decoding; and (4) ISE.FastSpec Greeks Decoding (collectively, the "Accused Decoding Instrumentalities" and with the ISE Accused Encoding Instrumentalities, the "Accused Instrumentalities").

In order for any of the accused instrumentalities to infringe, it must satisfy the limitations in each asserted claim.

A. Do ISE's FAST Applications "Analyze" or "Recognize" to Determine Data Type?

Each of the claims that ISE is accused of infringing require "analyzing" or "recognizing" to determine data type. ISE argues that Realtime relies on a "value check" performed by the instrumentality at issue to determine the data block type of data field type, and that this fails to comply with this Court's Markman opinion which requires "content categorization," not

checking a value. (See Mem. of Law in Support of ISE's Mot. for Summ. J. of Noninfringement ("ISE Mem.") (11 Civ. 6697, Dkt. No. 685) at 3-4.)

According to ISE, checking the value of a particular field is not the same as "analyzing" it to determine data type. ISE refers to the following example: if the COPY operator is being applied to a particular field, and the previous message had a "5" associated with the same field, a program implementing FAST's COPY operator will check to see if the value of the current field is "5"--but it will not try to determine data type. (See ISE Mem. at 4.) ISE argues that the FAST data types are already known and are set forth in templates prior to compression.

Realtime responds that the encoder is not simply checking a value. According to Realtime, even in the example that ISE provides, ISE is checking the value against the value in a prior message. Realtime argues that this is sufficient to constitute an "analysis." (Realtime Data, LLC's Mem. in Opp'n to ISE's Mot. for Summ. J. of Non-Infringement ("Opp'n Mot. 4") at 5-6.)

There is no dispute that there is a comparison of values that occurs as part of ISE's encoding process. The issue is whether the two-step process constitutes an "analysis."[8] (Notably, as presented, that argument is not based on the type

[8] The Court notes that the parties did not seek construction of the term "analyze" in connection with encoding with respect to the Markman opinion.

of content to which the "analysis" relates.) The parties have submitted dueling expert opinions on this question. The Court cannot resolve the question of fact as to whether ISE's two-step process constitutes an "analysis." Accordingly, the Court rejects this argument as a basis for a finding of summary judgment, and denies this aspect of Motion No. 4.

B. Are Fast Field Operators Encoders?

ISE argues that its Accused Instrumentalities do not employ encoders. ISE correctly notes that this Court has previously construed the term "encoder" to mean "hardware or software that compresses data by converting the contents of a data block (or data field) into a coded representation of those contents." See Realtime Data, 2012 WL 2394433, at *16. The Court's construction makes clear that a key distinction is that encoding must be more than simply "compression"--it must include some form of "coding" or changing of representation. See id. at *8-9. The Court also determined that throwing data "away" or "no change" does not constitute "encoding." Id. at *9.

Realtime has asserted that three of ISE's accused instrumentalities use or include encoders: COPY, DEFAULT and INCREMENT. According to ISE, none of those instrumentalities use or include encoders because none "code" data. (See ISE Mem. at 6.) Realtime asserts that in fact all three of the accused instrumentalities generate a coded representation. (See, e.g.,

Shamos Decl. ¶¶ 16, 17.) It relies upon the fact that ISE products all use FAST encoding techniques, as stated by Dr. Storer. (See Opp'n Mot. 4 (citing Storer Decl. Mot. 4 at 5-8).) According to Realtime, lossless encoding techniques can include "field encoding"--and field encoding includes copy, default and increment encoding. (Id. at 7-8.) Realtime also argues that such encoding techniques are content dependent--i.e., that default encoding is applied when the content of the data block or field has a data type "most common value"; "most common value" is determined based on prior knowledge of the data stream. (See id. at 7-8.)

Realtime asserts that all of ISE's accused products use copy encoding, citing Exhibit 2 of the Shamos Declaration. (Opp'n Mot. 4 at 8.) Exhibit 2 does not, however, provide this level of detail. (See Shamos Decl. Ex. 2.) Assuming some of the coding is copy encoding, that is when the content of a data block or field in the current message is analyzed by comparing the content with the content of the corresponding data block or field of a prior message. If two data types are the same, copy encoding is selected as the optimal encoding for the "redundant" data block type of the data block. (Shamos Decl. ¶¶ 17, 71, 73.) When copy encoding is utilized, the question is whether there has been any actual "encoding" at all. According to Dr. Shamos, there has been because a coded representation of the

data block is sent with the message in a PMAP that is included with the message. (See id. ¶¶ 17, 71, 73.) Dr. Shamos also states that some of the accused instrumentalities use increment encoding but he does not explain how. (See id. ¶ 17.)

Finally, according to Realtime, each of ISE's accused instrumentalities use "stop bit encoding" (also called "transfer encoding" or "variable byte encoding"). (Opp'n Mot. 4 at 9.) Stop bit encoding is used by ISE's accused instrumentalities when the data is not suitable for copy, default, or increment encoding. (Shamos Decl. ¶¶ 30, 35.) According to Dr. Shamos, this renders stop bit encoding a content independent compression technique. (See id. ¶ 35.) In addition, Dr. Shamos supports the proposition that "with stop bit encoding, the data field (or block) is converted into a coded representation comprising only the information content bits of the field (or block), which is transmitted along with one or more stop bits that indicate the unused bits of the data field that would not be transmitted." (Opp'n Mot. 4 at 9-10 (citing Shamos Decl. ¶¶ 19, 30, 35).) Thus, according to Dr. Shamos, the "stop bits" along with the information content bits of the encoded data together comprise a coded representation of the data block. (See Shamos Decl. ¶¶ 30, 35.)

ISE argues that stop bit encoding is not, in fact, content independent. (ISE's Reply Br. in Support of its Mot. for Summ.

J. of Noninfringement ("ISE Reply") (Dkt. No. 795) at 7-8.) ISE's System and Products Strategy Officer Maynard states, though, that stop bit encoding embeds intelligence into each byte in a field. (See Maynard Decl. ¶ 20.)

There is a material dispute of fact as to whether the Accused Instrumentalities employ encoders and if so, what type. Accordingly, summary judgment is denied on this basis.

C. Are ISE's Encoders 'Selected' During the Compression Process?

The third basis upon which ISE moves for summary judgment is that, according to ISE, all but two claims (claims 20 and 22 of the '568 Patent) require that an Accused Instrumentality "select" an encoder. (See ISE Mem. at 7.) According to ISE, none of its Accused Instrumentalities meet that requirement.

This Court has previously construed the phrase "selecting an encoder" as "choosing (or choose) an encoder (or lossless encoders) during the compression process based on analyses of content of the data blocks (or data fields)." See Realtime Data, 2012 WL 2394433, at *16. As the Court noted in its Markman opinion, the essential difference between the parties with respect to their proposed constructions has to do with the temporal moment when the encoder is selected. Id. at *12. The Court concluded that the selection occurs during the compression process and that this requires an analysis of the content of the data block. Id. at *13.

According to ISE, the selection of its encoders is preordained--i.e., the encoders are set forth in a template prior to the compression process. Thus, they cannot be selected during the compression process. (ISE Mem. at 8.) According to ISE's Maynard, the selection of the encoder could date back to March 2008 when the template was written. (Maynard Decl. ¶ 12; see also ISE Mem. at 8.)

Realtime disagrees. The essential difference between the parties is whether the encoder "selects" at the time of the creation of the template (which sets forth possible encoders), or whether the selection occurs when the data block is being analyzed during the compression process, when the template is referenced. According to Realtime, ISE's encoders are selected based on an analysis of the content of the data block. Part of that comparison involves an analysis of the data block against the template which sets forth possible encoders for certain data block types. (Opp'n Mot. 4 at 11.)

Realtime argues that "because ISE's section process occurs immediately after analysis, all of ISE's accused products contain the claim limitation." (Opp'n Mot. 4 at 12.) Realtime further argues that the fact that the template offers associations of data block types to optimal encoders does not replace the actual moment of selection of such encoder (even

using the template reference) during the compression process. (Id.)

The debate over the temporal moment of "selection" must be distinguished from other aspects of the selection of the encoder that are not a part of this particular argument for summary judgment. This particular argument is exclusively focused on when selection occurs, not how or whether the descriptor meets required limitations. In terms of this narrow argument, there is a material dispute of fact as to when the selection of the encoder occurs. Therefore, summary judgment is denied on this basis.

D. Does ISE Use a "Descriptor"?

ISE argues that each of the asserted claims, with the exception of claims 20 and 22 of the '568 Patent, requires that the accused instrumentality contain one or more "descriptors," wherein the descriptors indicate which lossless encoder has been selected. (See ISE Mem. at 8.)[9] ISE asserts that none of its Accused Instrumentalities meet this limitation. (Id.)

This Court has previously construed "descriptor with the encoded data which identifies" [the selected encoder], as "recognizable data that is appended to the encoded data for

---

[9] Claim 1 of the '568 Patent requires, for instance, "providing a descriptor with the encoded data which identifies the selected encoder." ('568 Patent col.23 l.44-45.)

specifying" the selected encoder. See Realtime Data, 2012 WL 2394433, at *16.

The facts relating to this descriptor issue are not in dispute; rather, their characterization is.

ISE argues that its accused instrumentalities do not have a descriptor. ISE contends to get around this issue Realtime tries to construct a "descriptor" through the combination of a PMAP and template ID or a message type. (See ISE Mem. at 9.) ISE argues that either combination does not identify the selected lossless encoders, as required. (Id.)

Dr. Shamos, however, argues that the combination of the PMAP and the template ID provide all of the information that is needed in order to determine which encoder was selected. (See Shamos Decl. ¶ 57.) Dr. Shamos does not, however, state that the information in the template ID or the PMAP in fact identify the selected encoder. Indeed, he could not because they do not.

The PMAP and template ID do not themselves identify the selected encoder; rather, they point to a fixed template which contains various encoder types. The PMAP and template ID then use that template to identify the encoder that would be used optimally to encode the data type at issue. There is no factual dispute that template is not "with" the encoded data; it is referenced by the PMAP and template ID. (See Storer Decl. Mot. 4 ¶ 151.) There is also no real "selection" of an encoder as

part of this process; the encoder is simply referenced from the template.

The critical point of distinction between the parties is whether or not the template can be considered to be "with" or attached to the encoded data. ISE's Maynard argues that it is not--the template is provided to the customer before the compression process even begins and all that is accompanying the data block itself is the template ID and PMAP. (Maynard Decl. ¶¶ 12-17.) Maynard states that the template ID and the PMAP do not themselves provide any indication as to which type of transfer encoding might have been applied to a given piece of data. (Id.)

Realtime argues that ISE has taken a far too narrow view of what a descriptor is--or can be. Realtime does not dispute that the template ID and the PMAP reference back to the template. (Opp'n Mot. 4 at 14.) According to Realtime, the PMAP and template ID are utilized at the decoding end to reference the template which will then indicate which encoder was used. (Id.)

The question on this motion is whether the template, which in fact identifies the encoder that has been used, is "with" the data block as required by the language of the claim. The answer is "no." Although the PMAP and template ID are in fact with the data block, they do not contain information regarding the selected encoder. It takes another step--away from the data

block--to determine which encoder has been selected. The "descriptor" therefore requires both information that is "with" the data block and information that is "not with" the data block. The claim requires that the descriptor be with the data block.

Accordingly, ISE is entitled to summary judgment on this basis.

E. Is ISE's Transfer Encoding Content Independent Data Compression?

ISE argues that claims 22, 25, 26, 29, 34 and 35 of the '651 Patent, claims 14 and 19 of the '747 Patent, and claim 22 of the '568 Patent all require that in order to be infringed, the accused instrumentality must utilize content independent data compression. (ISE Mem. at 10.) According to ISE, none of its Accused Instrumentalities meet this requirement.

This Court has previously construed "content independent data compression" to mean

> compression that is applied to input data that is not compressed with content dependent data compression, the compression applied using one or more encoders without regard to the encoder's (or encoders') ability to effectively encoded the data block type (or data field type).

Realtime Data, 2012 WL 2394433, at *16.

According to ISE, Realtime's infringement contentions (and Dr. Shamos) rely upon stop bit encoding to meet the content independent data compression requirement. (ISE Mem. at 11.)

ISE argues that stop bit encoding--or transfer encoding--is only used on certain data types and therefore must be content dependent. (Id.) According to ISE's Maynard, ISE chooses whether to apply transfer encoding based on the type of data in a field and whether it will be able effectively to encode that data. (Maynard Decl. ¶¶ 27-28.)

Realtime disagrees. According to Realtime, transfer encoding--which ISE concedes it uses--is used when three other types of encoders are not optimal (copy, incremental, and default encoders); that does not mean that the content of the data block is known. In fact, according to Realtime, the content of the data block can nonetheless be one of several different types including fields containing integer numbers, signed integers, unsigned integers, scaled numbers and ASCII strings. (Opp'n Mot. 4 at 16.) Transfer encoding may be applied to other types of data in addition to these but its application may not be optimal. (Id.)

Whether transfer encoding, which is used when other encoding techniques are not optimal, but when the content can be one of a number of different types, is content independent is a question of fact for the jury. Accordingly, the Court denies summary judgment on this basis.

F. Is Stop Bit Encoding "Lossless"?

ISE argues that all of the asserted claims of the '651 and '747 Patents require that the accused instrumentality use a lossless encoder or lossless decoder. (ISE Mem. at 11.)

This Court previously construed "lossless" to mean "technique, software or hardware that fully preserves the original unencoded data such that the decoded data is identical to the unencoded data." Realtime Data, 2012 WL 2394433, at *16.

ISE argues that Realtime has consistently cited ISE's stop bit encoding feature of FAST transfer encoding as meeting the lossless requirement. (ISE Mem. at 11-12.) As a factual matter, ISE argues that Realtime has never shown that any customers who decode ISE's encoded market data get data that is identical to that which was encoded. (Id. at 12.) According to ISE, that amounts to a failure of proof--i.e., that whether the stop bit encoding is lossless or not cannot be demonstrated on the current record and therefore summary judgment is warranted. (Id.)

ISE further argues that even if Realtime had undertaken to show that stop bit encoding was lossless, it could not have. (See ISE Mem. at 12.) ISE's Maynard states that when many feeds are FAST encoded in ISE's system, data is lost. (Maynard Decl. ¶¶ 22-26.) Moreover, when ISE decodes OPRA feeds, that decoding

is not lossless because the encoding was not lossless. (ISE Mem. at 13.)

Realtime first argues that it has not limited itself to ISE's stop bit encoding, and that ISE's accused products, which include but are not limited to stop bit encoding, are lossless. Realtime points to an analysis of source code conducted by Dr. Shamos. (Opp'n Mot. 4 at 17.) According to Realtime, although there may be a lack of total identicality during the transmission process, once the data is encoded losslessly it can be decoded losslessly. (Id.)

Realtime points to ISE's transfer, copy, increment, and default encoding techniques as lossless. Realtime argues that the use of the PMAP and template ID allow for the reconstruction at the decoding end of identical data. Thus, while the data may be transmitted with less than all the data, the decoding process can achieve the required identicality. (Opp'n Mot. 4 at 17.) Realtime is correct that under the Court's construction, identicality is judged as the point of decoding. See Realtime Data, 2012 WL 2394433, at *16.

There is a material issue of fact, however, as to whether ISE's encoding techniques are lossless. Although it may be that there is no record of a before-and-after demonstration of encoding and decoding customer data, given Dr. Shamos' expertise in the field, he can opine as to his views, the jury can credit

his testimony (or not) and weigh that testimony to whatever extent it believes it deserves in light of the analysis Dr. Shamos has (or has) not conducted.

The Court denies summary judgment on this basis.

G. Is there a Data Stream at the Point of Encoding?

ISE argues that all of the asserted claims in the '568 Patent require that the accused encoder must operate on a "data stream." (ISE Mem. at 13.) ISE argues that since none of its Accused Instrumentalities perform this step, none infringe. (Id.)

This Court previously construed "data stream" to mean "one or more blocks transmitted in sequence from an external source whose characteristics are not controlled by the data encoder or decoder." Realtime Data, 2012 WL 2394433, at *16.

ISE's Maynard states that for every ISE Accused Instrumentality, ISE generates its feed internally from stored data. (Maynard Decl. ¶¶ 41-43.) Thus, according to ISE, since the data feeds are "not external," they fail to meet one of the construed requirements for a data stream. (ISE Mem. at 13.) ISE points to statements by Dr. Shamos in which he appears to indicate that "ISE feeds" are internal (and therefore not external) to ISE. (Id. at 13-14 (citing Shamos Decl. ¶¶ 45-46).) ISE's Maynard states, however, that ISE's Accused Encoding Instrumentalities encode data that comes from within

ISE's own systems, out of its own trading engines. (Maynard Decl. ¶ 41.) ISE's trading data is always under its own control. (Id.)

According to Realtime, ISE is simply wrong that the trading data is internal. Realtime argues that in fact trading data necessarily derives from external sources: the data from trades in markets or other institutions that are not part of ISE. ISE replies that Dr. Shamos admitted that he had no idea where ISE's trading data came from, and the only evidence in the record is from Maynard who states that it comes entirely from ISE's own internal trading engine.

According to Maynard, ISE's Accused Encoding Instrumentalities encode data that come from within ISE's own systems, and is passed from the trading engine to the server which performs the FAST encoding; that server stores the trading data in memory to create a duplicate set of trading data; the FAST encoder then uses this stored data in the memory of the server to, inter alia, form the FAST feeds. (Maynard Decl. ¶ 41.)

Realtime argues that ISE misconstrues the meaning of "external." (See Opp'n Mot. 4 at 20-21.) Realtime references certain prior art--"Sebastian" (cited in the reexamination proceeding for the '568 Patent)--to show a contrast between encoders that do and do not control change of sequence of data

blocks. (Id. at 21.) In Sebastian, an encoder sorts incoming data blocks into buckets and compresses the data blocks in a particular order; thus, the data stream is not "external" to the encoder. (Id.) Realtime argues that ISE's Accused Instrumentalities do apply the FAST protocol to the data blocks as the data blocks are received and do not control or change the characteristics of the data blocks. (Id.)

The Court agrees that there is no material issue of fact with respect to whether the data is external. The evidentiary record is one sided on this point: the data comes from ISE's own trading engine. Although Realtime has asserted that this must mean that the trades reported in the data originated externally, that is an unsupported factual assertion. At the summary judgment stage, more is required. Thus, there is no material issue of fact that ISE's Encoding Instrumentalities do not infringe claims 15, 20, 22 and 32 of the '568 Patent. Summary judgment is granted on this issue.

H. Does ISE's ISE.FastProcessing Encoding Produce an Output Data Stream?

ISE argues that the encoding claims of both the '651 Patent (claims 13, 18, 19, 21, 22, 25, 26, 29, 34, 35, 43, 47 and 49) and the '747 Patent (claims 14, 19) all require that the result or output of encoding be a data stream. (ISE Mem. at 15.) ISE argues that the record is devoid of any evidence that this requirement is met with respect to the ISE.FastProcessing

library. (See id.) According to ISE's Maynard, this library is used exclusively for encoding and decoding journal files, not for creating or decoding data streams. (See Maynard Decl. ¶¶ 35-37.) Journal records are maintained for audit purposes. (Id. ¶ 42.) ISE argues that its Journal Record Application cannot constitute the required "external source"--and that since the journal records are never sent to a third party they are not part of any output data stream. (ISE Mem. at 16.)

Realtime does not dispute that the ISE.FastProcessing Encoding creates journal files. However, according to Realtime, it is of no moment that those journal files are never sent to third parties--there is no requirement that they terminate with a third party. (Opp'n Mot. 4 at 22.) Again, Realtime relies upon its assertion, without record support and as contradicted by Maynard, that the information that forms the trading files must come from some external source. As stated, Maynard is clear that the trading files are generated from ISE's own trading engine and do not come from an external source. (See Maynard Decl. ¶ 41.) Realtime depends upon this assertion to make the following statement:

> Because the activity in its trading engine must be related to information on trades, which are performed externally to ISE's Accused Encoders, ISE's journal records must contain data which was transmitted by an external source. Therefore, ISE's Accused ISE.FastProcessing encoding produces an output 'data stream.'

(Opp'n Mot. 4 at 22.) Realtime does not support its assertion with any citation to any evidence--even an expert opinion. It is mere speculation.

Accordingly, the Court finds that there is no triable issue of fact that the accused ISE.FastProcessing encoding does not infringe the claims recited above. Summary judgment is granted in this regard.

I. <u>Are There Data Packets At the Time of Encoding?</u>

ISE argues that claims 1 (unasserted), 15, and 32 of the '568 Patent require that in order to be infringed, the encoder must operate on data in a "data packet." (ISE Mem. at 16.) ISE claims that it does not in fact encode data packets and therefore cannot infringe claims 1, 15, and 32 of the '568 Patent. (ISE Mem. at 10.)

The parties agree that the term "packet" means "information limited in type, format and content and able to be transmitted as a unit across a packet-switched network, the packet including control information that enables the packet to be delivered to an intended destination in the network." (ISE Mem. at 16.)

ISE asserts that its Encoding Instrumentalities, namely ISE.Fast Encoding, ISE.FastProcessing Encoding and MIDAS Encoding, do not FAST-encode data that is in a data packet. (ISE Mem. at 16.) According to ISE's Maynard, trading data may arrive in packets to the server, but they are stored in memory

and are no longer "in packets" after that. (ISE Reply at 10; Maynard Decl. ¶ 41.) According to ISE, there is no evidence that ISE encodes data in packets at all--and indeed, Maynard states that it does not. (Maynard Decl. ¶ 41.)

Realtime argues that because the trading data is of a specific type (numerical) and format (a format acceptable to the server for which it was intended) and content (records of trades) is transmitted "over some kind of network from the trading engine to a server." (Opp'n Mot. 4 at 23.) Realtime extrapolates from those assertions that this must mean that the data meets the agreed definition of packets and travels in packets in the data stream. This is not supported by any facts. It is unclear that the format is limited in type--Realtime simply asserts without support that it is. In addition, there is nothing supportive of the proposition that the data is being transmitted as a unit, or over a packet-switched network (versus, as Realtime states, some kind of network from the trading engine). (Opp'n Mot. 4 at 23.)

Realtime's arguments do not raise a triable issue as to whether the data is in packets when it enters the data stream. Thus, ISE is entitled to summary judgment as to claims 15 and 32 of the '568 Patent.

J. Does ISE Infringe Decoder Claims If It Does Not Perform the Encoding Step?

ISE asserts that its Accused Decoding Instrumentalities, ISE's Exegy OPRA Decoding, OPRA Decoding and ISE.FastSpec Greeks Decoding cannot infringe claims 95, 97, 108, and 112 of the '651 Patent because they do not perform the required encoding step. (ISE Mem. at 17.)

Each of these decoding claims contains the following limitation: "wherein the lossless encoders are selected based on analyses of content of the data blocks [or fields]." (Id.) The parties have agreed that this phrase as used in these claims means "the system (or method) selects the lossless encoders based on analyses of content of the data blocks (or data fields)." (Id.)

According to ISE, Dr. Shamos concedes that these decoding claims require encoding. (ISE Mem. at 17 (citing Shamos Decl. ¶ 78).) The critical issue is whether to infringe these claims a single party--here, ISE--must itself perform (or directly control another who performs) both the encoding and decoding steps.

ISE argues that there is no evidentiary support--and no proffered opinion by Dr. Shamos--that ISE performs both the encoding and decoding steps for these claims. (ISE Mem. at 18.) Indeed, it cites to Maynard as stating the opposite. (Id.) According to Maynard, OPRA sends a feed to ISE, which OPRA

itself encoded; ISE then uses its OPRA decoding software to decode that feed--but it had nothing to do with the original encoding. (See Maynard Decl. ¶ 44.) There is no argument that ISE controls OPRA.

With respect to the Greeks decoding function, the Volera Greeks feed comes from an unrelated third party company called Hanweek Associates LLC ("Hanweek"). ISE does own 20 percent of Hanweek.

Realtime does not dispute ISE's factual arguments. Instead, it simply asserts that ISE "was the mastermind" behind the performance of all of the steps of these asserted claims. (Opp'n Mot. 4 at 24.) Such an unsupported statement is insufficient at the summary judgment stage.

There is no material issue of fact with respect to whether ISE's OPRA decoder infringes--it does not; it does not perform an essential step of the claim. Accordingly, summary judgment is granted as to these claims.

K. Does ISE's MIDAS Encoding Infringe Claims 20 and 22?

ISE argues that claims 20 and 22 of the '568 Patent require the step of "processing the description file with a data compression compiler, and outputting an executable file that is used to process a stream of data by recognizing data field types in the data stream and applying encoders associated with the

recognized data field types to encode the data stream." (ISE Mem. at 19.)

According to ISE, its accused MIDAS Encoding is based on software obtained from a third party, OMX. (Id. at 19.) ISE's Maynard states that ISE has never possessed the source code for this software and has never compiled it. (See Maynard Decl. ¶¶ 33, 45.) Realtime does not contradict these statements.

Instead, Realtime argues that "ISE contracted with OMX for 'deliver[y] of a production-ready version of the Licensed Product.'" (Opp'n Mot. 4 at 23.) Realtime then relies upon the legal proposition that one cannot avoid infringement by having another perform one of the required steps. (See id. at 24.) That argument is applicable to situations in which another is contractually obligated to perform the claimed steps. See Akamai Techs., Inc. v. Limelight Networks, Inc., 629 F.3d 1311 (Fed. Cir. 2010). Critically, there is more to the claimed step than what can be read as a matter of law or assumed as a matter of fact into the delivery of a production-ready version of the software: the steps require processing the description file with a data compression compiler, and outputting an executable file that is used to process a stream of data by recognizing data field types. (See ISE Reply at 10; '568 Patent col.24 ll.56-67.)

Realtime has not proffered the opinion of an expert that delivery of the software requires compiling; or that any of these other steps are a necessary part of such delivery. Accordingly, there is no triable issue of fact on this issue and ISE is entitled to summary judgment with respect to claims 20 and 22 of the '568 Patent.

Conclusion: Motion No. 4

The above rulings complete the Court's decision on ISE's motion for summary judgment of noninfringement (Motion No. 4). ISE's motion for summary judgment is GRANTED IN PART and DENIED IN PART as set forth above.

VI. MOTION 5

Defendants CME Group, Inc., Board of Trade of the City of Chicago, Inc., and the New York Mercantile Exchange, Inc. (collectively the "CME Defendants") move for summary judgment on the basis that their encoding and decoding systems and methods do not analyze the content of a field or block to determine a data field type or data block type. (Mem. of Law in Support of CME Defs.' Mot. for Summ. J. of Noninfringement Based on Lack of Determining a Data Block or Field Type ("CME Mem.") (Dkt. No. 660) at 1.)[10]

[10] The following defendants join in Motion No. 5: The Goldman Sachs Group, Inc., Goldman, Sachs & Co., Goldman Sachs Execution & Clearing, L.P., J.P. Morgan Chase & Co., J.P. Morgan Securities, Inc., J.P. Morgan Clearing Corp., Morgan Stanley, Morgan Stanley & Co, Incorporated, HSBC Bank USA, National Association, HSBC Securities USA, Inc., Credit Suisse Holdings (USA), Inc., Credit Suisse Securities (USA), LLC, Thomson Reuters Corporation, Bloomberg

In its infringement contentions with respect to CME, Realtime asserted that the claim requirement of determining a data type was met because the encoder "checks" a current value of a data field. (CME Mem. at 2.) This Court previously construed data block type and data field type as "categorization of the data in the field (or block) as one of ASCII, image data, multimedia data, signed and unsigned integers, pointers or other data types." Realtime Data, 2012 WL 2394433, at *16.

The crux of the CME Defendants' arguments in support of their motion is that prior to the Markman opinion, and then in connection with its infringement contentions, Realtime has used the concept of "checking a value" rather than categorization of the data. According to the CME Defendants, Realtime has used this language relating to "checking a value" with respect to 7 of the 8 asserted independent claims, and has used similar language for the eighth: claims 13, 22, 29, 60 of the '651 Patent, claims 1 and 20 of the '568 Patent, and claims 14 and 19 of the '747 Patent. (CME Mem. at 6.)[11] They argue that Realtime is wrong in asserting that the requirement of analyzing the "content of the data block to determine data type" is met when

L.P., Interactive Data Corporation, BATS Trading, Inc., BATS Exchange, Inc., The NASDAQ OMX Group, NASDAQ OMX PHLX LLC, Chicago Board Options Exchange, Incorporated, BOX Options Exchange LLC, FactSet, Inc., and International Securities Exchange.

[11] The law is clear that if independent claims are not infringed, neither are dependent claims. See Minn. Min. & Mfg. Co. v. Chemque, Inc., 303 F.3d 1294, 1300 (Fed. Cir. 2002).

the encoder checks for a relation between the current value of the message field and a value of the corresponding message field in a previous message in the data packet because the Court explicitly reject this construction of the claim in the Markman opinion. (Id. at 6-7.)

Realtime argues that the CME Defendants truncate the steps they are outlining and that they are undertaking a two-step, analytical process: checking a value and then comparing that value. (Realtime Data, LLC's Mem. in Opp'n to CME Defs.' Mot. for Summ. J. of Noninfringement Based on the Lack of Determining a Data Block or Field Type ("Opp'n No. 5") at 9-11.)[12] The CME Defendants respond that this may be the "analysis" step--but it does not answer whether the CME Defendants determine a data block type or data field type during the compression as required by the claim; and the CME Defendants argue that this determining step is more than checking a value. (See Reply in Support of Defs.' Mot. for Summ. J. of Noninfringement Based on Lack of Determining a Data Block or Field Type (11 Civ. 6697, Dkt. No. 800) at 3.)

Realtime grounds its response in the specification language of the patents-in-suit. According to Realtime, a "data type" may be a characteristic of the data block. (Opp'n Mot. 5 at 3.)

[12] ISE's motion for summary judgment based on whether an "analysis" occurred was denied, see Part V.4.A. supra; ISE did not move based on content categorization.

It is clear from Realtime's description of how it performs this step, that it is performing a value check. The question is whether there is an issue of fact as to where this value check can constitute content categorization. There is not. The Court resolved this issue in connection with its Markman opinion.

Realtime argues the CME Defendants' copy encoding technique determines whether the data block is redundant of the corresponding data block in the prior message and that they perform this step through a "content check." (Opp'n Mot. 5 at 4.) This is, as Dr. Storer states (uncontradicted by Dr. Shamos), really a value check--not a content categorization (e.g., determining whether the content is ASCII, multimedia, signed or unsigned integers, etc.).

With respect to its increment encoding technique, Realtime states that the CME Defendants determine the data field (or block) type by analyzing whether the data block has a sequential difference of one. (Id. at 4.) Realtime argues that this sequential difference is also a characteristic of the data block. This is a value check.

Finally, with respect to the CME Defendants' default encoding technique, Realtime asserts that the data field (block) type is also determined by identifying whether the data block has content that is the most common value for that data block or field that is expected for that data block based on a priori

knowledge of the data stream. (Id. at 4.) This is also clearly a species of checking a value and not of content categorization.[13]

The mere repetition of the word "content" throughout Realtime's argument cannot change what the experts opine is occurring: the CME Defendants' encoding techniques check values; they are not examining the content of the data block for its proper categorization.

The CME Defendants are entitled to summary judgment with respect to their Accused Encoding Instrumentalities. Their motion is GRANTED.

VII. MOTION 6

Defendants NYSE Euronext, NYSE ARCA, Inc., NYSE AMEX, LLC and Securities Industry Automation Corporation (collectively "NYSE"), and OPRA (collectively with NYSE, the "NYSE/OPRA Defendants") have moved for summary judgment of non-infringement as to all claims of the patents-in-suit.[14]

According to the NYSE/OPRA Defendants, the encoding and data block type (or field) limitations, which are asserted in

[13] As an alternative argument, the CME Defendants assert that they does not infringe because it does not meet the "data block type" or "data field type" limitation of the claims. Here again, the question is whether checking a value is determining a categorization of a data block (or field) type. As stated above, the answer to the question posed in this way (which is only slightly different from that discussed above) is "no." Checking a value is not the same as categorizing data type as, inter alia, ASCII, multimedia, signed and unsigned integers, etc. This is an alternative basis for granting summary judgment.

[14] The same defendants that joined in Motion No. 5 similarly join in Motion No. 6.

every claim they are accused of infringing, and the selecting limitation, which is asserted in all but two claims asserted against them (claims 20 and 22 of the '568 Patent), are absent from the accused systems. (See NYSE and OPRA Defs.' Mem. of Law in Support of Mot. for Summ. J. of Noninfringement Based on the Absence of Encoding, Data Block (or Field) Type, and Selecting Limitations ("NYSE/OPRA Mem.") at 1.)

The NYSE/OPRA Defendants make two arguments with regard to encoding: (1) that they do not encode since the NYSE and OPRA systems all throw data away and result in no change, which this Court rejected as constituting "encoding" in its Markman opinion; and (2) that the field operators do not generate a coded representation of the incoming data as this Court in its Markman opinion stated was required in order to encode.

As to the data type limitation, the NYSE/OPRA Defendants repeat the arguments made by the CME Defendants in Motion 5 above: that there is no analysis to determine the data type nor is there an analysis of data type in connection with the selection of the encoder. (NYSE/OPRA Mem. at 2.)

In terms of the selecting limitation, the NYSE/OPRA Defendants argue that this is made during the operation of the field operator identified by Realtime as a content dependent encoder. In contrast, the patent covers selection of the encoder itself, which necessarily has to begin before any

encoder begins encoding. In addition, according to the NYSE/OPRA Defendants, the two encoders identified by Realtime are not alternatives--they are in fact applied one after the other. Finally, the NYSE/OPRA Defendants argue that the selection identified in the accused products is based on data value, not data type as the claims require. (Id. at 2.)

A. Analyzing the Content of the Data Block Type

The NYSE/OPRA Defendants make the same arguments as those made by the CME Defendants with respect to the data block (or field) type limitation. They argue that a value check is not the type of data categorization which must occur. Realtime argues that it is, because the “value” is a characteristic of the data, and constitutes “other data type” even under the Court’s Markman ruling. Realtime also argues that the NYSE/OPRA Defendants undertake a two step process that does constitute the necessary analysis of the data block (or field) type.

The Court refers to its analysis of these same arguments with respect to Motion 5. The NYSE/OPRA Defendants are correct that there is no triable issue of fact with respect to their data block (or field) type arguments, and they are entitled to summary judgment on this basis as a matter of law.

B. The Encoding Limitation

According to Realtime, the NYSE/OPRA Defendants’ field operators do, in fact, convert the contents of a data block (or

data field) into a coded representation of those blocks. The NYSE/OPRA Defendants use copy, default, increment and transfer (or stop bit) encoding techniques. For the same reasons set forth in connection with ISE's motion for summary judgment on this issue (Motion No. 4), there are material issues of fact that preclude summary judgment.

Specifically, according to Realtime's expert, Dr. Shamos, in fact all three of these field operators generate a coded representation. (See Shamos Decl. ¶¶ 17, 40, Ex. 1.) Realtime asserts that these products all use FAST encoding techniques. (Realtime Data, LLC's Mem. in Opp'n to NYSE & OPRA Defs.' Mot. for Summ. J. of Noninfringement Based on the Absence of Encoding, Data Block (or Field) Type, and Selecting Limitations ("Opp'n Mot. 6") at 13.) According to Dr. Shamos, lossless encoding techniques can include "field encoding"--and field encoding includes copy, default, and increment encoding. (Shamos Decl. ¶¶ 17, 30.)

With respect to stop bit encoding, Drs. Shamos and Storer again disagree. According to Dr. Shamos, while data bits may be thrown away in stop bit encoding, and transmitted in that manner, they can be identically replicated at the decoding end. (Shamos Decl. ¶¶ 19, 30, 35.) Dr. Storer opines that what Realtime calls an encoder is not. (Storer Decl. Mot. 6 ¶ 43.)

For the reasons stated in connection with ISE's motion on this same issue, the Court finds there are triable issues of fact as to whether the encoders in fact "code." Accordingly, the motion is denied on these grounds.

C. The Selecting Limitation

The NYSE/OPRA Defendants claim that the selection limitation cannot be met because it occurs far earlier than "during the compression process" as required. According to Dr. Storer, the selection of an encoder occurs at the time the code for the template is written. (See NYSE/OPRA Mem. at 16 (citing Storer Decl. ¶¶ 21, 23, 29).)

As the Court noted in its Markman opinion, the essential difference between the parties with respect to their proposed constructions has to do with the temporal moment when the encoder is selected. Realtime Data, 2012 WL 2394433, at *12. The Court concluded that the selection occurs during the compression process and that this requires an analysis of the content of the data block. Id. at *13.

ISE made a somewhat similar argument to that being made by the NYSE/OPRA Defendants. In both instances, the question is whether the selection of the encoder occurs during or prior to the compression process. With respect to ISE, the presence of the PMAP and template ID raised a question of fact.

Here, the question of fact is based on Dr. Shamos' contention that the FAST copy encoding technique actually requires directly analyzing the content of each data block (or data field) to be transmitted; this occurs as the data blocks are present during the compression process, not when the code was written. The Court cannot therefore grant summary judgment on the basis of this argument.

Even if the selection occurs during the compression process, the NYSE/OPRA Defendants argue that the selection limitation is still not met because that selection is never based on data type. (See NYSE/OPRA Mem. at 16-17.) Realtime disagrees. (See Opp'n No. 6 at 20.) Realtime's response on this issue is, however, less than comprehensible. It appears that it is arguing that there is an analysis of data block/field to determine data type and to select an encoder; however, if this is the same type of analysis that is simply checking a value (it is unclear from Realtime's memorandum and supporting declaration), then this is not the type of data block analysis that is required.

The Court assumes that this is in fact what Realtime is arguing and therefore grants summary judgment based on this argument. If the Court has misconstrued Realtime's argument, then it should clarify its position promptly.

VIII. MOTION 8

A group of defendants (the "Motion 8 Defendants")[15] have moved for summary judgment on largely the same basis as set forth in Motion 4(D): that a descriptor is not "with" the data blocks. The Motion 8 Defendants argue that independent claims 14 and 19 of the '747 Patent, claims 13, 22, 29, 43, 91 and 108 of the '651 Patent, and claims 15 and 32 of the '568 Patent (and the claims that depend from these claims) all require descriptors.[16] (See Defs.' Mem. of Law in Support of Motion for Summ. J. of Non-Infringement: Descriptor Limitation ("Mot. 8 Mem.") at 1.)

Here, the Motion 8 Defendants do not argue that the "PMAP plus template ID, referring to an encoder" is not a "descriptor" as ISE does, but rather argue that this combination does not "specify" the encoder as the claims require, at most there is an indirect reference. Those semantic differences aside, the

[15] The Motion 8 Defendants are NYSE Euronext, NYSE ARCA, Inc., NYSE AMEX, LLC, Securities Industry Automation Corporation, OPRA, HSBC Bank USA, N.A., HSBC Securities (USA), Inc., Chicago Board Options Exchange, Incorporated, FactSet Research Systems, Inc, BATS Trading, Inc., BATS Exchange, Inc., Morgan Stanley, Morgan Stanley & Co. Inc., The Goldman Sachs Group, Goldman, Sachs, & Co., Goldman Sachs Execution & Clearing, L.P., J.P. Morgan Chase & Co., J.P. Morgan Securities Inc., J.P. Morgan Clearing Corp., Thomson Reuters Corp., BOX Options Exchange LLC, International Securities Exchange LLC, The NASDAQ OMX Group, Inc., NASDAQ OMX PHLX LLC, Bloomberg, L.P., CME Group Inc., Board of Trade of the City of Chicago, Inc., New York Mercantile Exchange, Inc., Penson Worldwide, Inc., and Nexa Technologies, Inc.

[16] As an example, claim 14 of the '747 Patent requires "providing a descriptor for the compressed data packet in the data stream, wherein the descriptor indicates the one or more selected lossless encoders for the encoded data block."

Motion 8 Defendants' arguments and ISE's (see Part V.D. supra) reduce to the same and thus, are resolved the same way.

This Court construed "descriptor[s] indicate" as the equivalent of "descriptor with . . .": both require that "recognizable data that is appended to the encoded data for specifying". See Realtime Data, 2012 WL 2394433, at *16. The Court specifically analyzed the question of whether the descriptor must be physically attached to the data or can be associated with the data. It determined that it must be attached in the sense of being "with." Thus, the Court rejected the construction that would allow the referenced template, which is associated with the data. The Court explained that its construction was consistent with multiple references in the specification itself which used language such as "with" or "appended". As this Court found, "[T]here is no support in the specification or claims for the descriptor to be completely detached from the data block." Realtime, 2012 WL 2394433, at *15. The requirement that the template ID and PMAP reference a detached template in order to determine which encoder has been used, does not meet the claim limitation.

Accordingly, the Motion 8 Defendants are entitled to summary judgment on this basis.

IX. MOTION 9

Defendants Credit Suisse Holdings (USA), Inc. and Credit Suisse Securities (USA) LLC (collectively, "Credit Suisse") have also moved for summary judgment based on the descriptor limitation.[17] Credit Suisse's description of the process is somewhat different from that described by in Motions 4 and 8, but nonetheless makes the same major points.

In Credit Suisse's motion, it is clear(er) that the template may be sent to the decoding system in advance. In Motions 4 and 8 it was unclear where the template physically resided, but it was clear that it did not reside with the data block which had the template ID and PMAP. Credit Suisse states that "the decoding system receives the Template in advance, and stores a copy of it. Subsequently, upon receipt of an encoded FAST message, this copy of the Template is used to determine which encoders were used to encode the message." (See Credit Suisse's Mem. in Support of Mot. for Summ. J. of Noninfringement: Descriptor Limitation ("CS Mem.") at 3.) As a matter of undisputed fact, the template is not "with" or appended to the data block.

[17] The defendants who joined in the motion: HSBC Bank USA, N.A., HSBC Securities (USA), Inc., Chicago Board Options Exchange, Incorporated, BATS Trading, Inc., BATS Exchange, Inc., Morgan Stanley, Morgan Stanley & Co. Inc., The Goldman Sachs Group, Goldman, Sachs, & Co., Goldman Sachs Execution & Clearing, L.P., J.P. Morgan Chase & Co., J.P. Morgan Securities Inc., J.P. Morgan Clearing Corp., Thomson Reuters Corp., BOX Options Exchange LLC, International Securities Exchange LLC, The NASDAQ OMX Group, Inc., NASDAQ OMX PHLX LLC, Bloomberg, L.P., Interactive Data Corp., CME Group Inc., Board of Trade of the City of Chicago, Inc., and New York Mercantile Exchange, Inc..

It is still a reference tool that is separate and apart from the data block that has been encoded. Accordingly, this process does not meet the required limitations of the claims requiring a descriptor, and summary judgment is therefore granted.

CONCLUSION

The Court's rulings on Motions 4, 5, 6, 8 and 9 are as set forth above.

Motion No. 4 is GRANTED IN PART and DENIED IN PART. Motion No. 5 is GRANTED. Motion No. 6 is GRANTED IN PART and DENIED IN PART. Motion No. 8 is GRANTED. Motion No. 9 is GRANTED.

The parties are directed to submit to the Court a letter within four business days of the issuance of this order setting forth (1) whether and how these rulings affect the upcoming trial in the Exchange Action; and (2) the order that the Court should proceed to resolve the remaining motions for summary judgment.

The Clerk of the Court is directed to terminate the following motions:

11 Civ. 6696: Dkt. No. 523

11 Civ. 6697: Dkt. Nos. 684, 659, 680, 687

11 Civ. 6699: Dkt. Nos. 92, 109, 112

11 Civ. 6701: Dkt. No. 111

11 Civ. 6702: Dkt. Nos. 132, 149, 152

11 Civ. 6704: Dkt. Nos. 114

SO ORDERED:

Dated: New York, New York
November 15, 2012

K B. Forrest

KATHERINE B. FORREST
United States District Judge

US007417568B2

(12) **United States Patent**
Fallon et al.

(10) **Patent No.: US 7,417,568 B2**
(45) **Date of Patent: Aug. 26, 2008**

(54) **SYSTEM AND METHOD FOR DATA FEED ACCELERATION AND ENCRYPTION**

(75) Inventors: **James J. Fallon**, Armonk, NY (US); **Paul F. Pickel**, Bethpage, NY (US); **Stephen J. McErlain**, New York, NY (US); **Carlton J. Melone**, Ridgewood, NJ (US)

(73) Assignee: **Realtime Data LLC**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 527 days.

(21) Appl. No.: **10/434,305**

(22) Filed: **May 7, 2003**

(65) **Prior Publication Data**

US 2004/0042506 A1 Mar. 4, 2004

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/969,987, filed on Oct. 3, 2001, said application No. 60/237,571 application No. 10/434,305.

(60) Provisional application No. 60/378,517, filed on May 7, 2002.

(51) **Int. Cl.**
H03M 7/38 (2006.01)

(52) **U.S. Cl.** **341/51**; 341/50; 370/355; 345/419

(58) **Field of Classification Search** 341/51, 341/50; 370/355; 380/212; 709/238; 358/400; 345/419; 714/752
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,127,518 A 11/1978 Coy et al.
4,302,775 A 11/1981 Widergren et al.
4,394,774 A 7/1983 Widergren et al.
4,574,351 A 3/1986 Dang et al.
4,593,324 A 6/1986 Ohkubo et al.
4,682,150 A 7/1987 Mathes et al.
4,730,348 A 3/1988 MacCrisken
4,804,959 A 2/1989 Makansi et al.
4,870,415 A 9/1989 Van Maren et al.
4,872,009 A 10/1989 Tsukiyama et al.
4,876,541 A 10/1989 Storer
4,888,812 A 12/1989 Dinan et al.
4,906,995 A 3/1990 Swanson

(Continued)

FOREIGN PATENT DOCUMENTS

DE 4127518 2/1992

(Continued)

OTHER PUBLICATIONS

Jack Venbrux, *A VLSI Chip Set for High-Speed Lossless Data Compression*, IEEE Trans. On Circuits and Systems for Video Technology, vol. 2, No. 4, Dec. 1992, pp. 381-391.

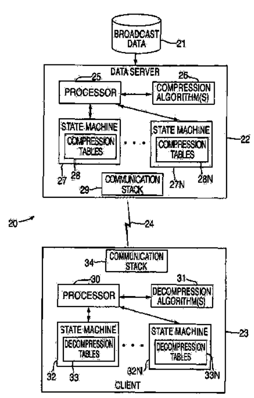
(Continued)

Primary Examiner—Jean B Jeanglaude
(74) *Attorney, Agent, or Firm*—Ropes & Gray LLP

(57) **ABSTRACT**

Systems and methods for providing accelerated transmission of broadcast data, such as financial data and news feeds, over a communication channel using data compression and decompression to provide secure transmission and transparent multiplication of communication bandwidth, as well as reduce the latency associated with data transmission of conventional systems.

70 Claims, 6 Drawing Sheets



U.S. PATENT DOCUMENTS

4,929,946	A	5/1990	O'Brien et al.	
4,965,675	A	10/1990	Hori et al.	
4,988,998	A	1/1991	O'Brien	341/55
5,028,922	A	7/1991	Huang	
5,045,848	A	9/1991	Fascenda	
5,045,852	A	9/1991	Mitchell et al.	
5,046,027	A	9/1991	Taaffe et al.	
5,049,881	A	9/1991	Gibson et al.	
5,091,782	A	2/1992	Krause et al.	348/400.1
5,097,261	A	3/1992	Langdon, Jr. et al.	
5,113,522	A	5/1992	Dinwiddie, Jr. et al.	
5,121,342	A	6/1992	Szymborski	
5,150,430	A	9/1992	Chu	
5,159,336	A	10/1992	Rabin et al.	
5,175,543	A	12/1992	Lantz	
5,179,651	A	1/1993	Taaffe et al.	
5,187,793	A	2/1993	Keith et al.	
5,191,431	A	3/1993	Hasegawa et al.	
5,204,756	A	4/1993	Chevion et al.	
5,209,220	A	5/1993	Hiyama et al.	
5,212,742	A	5/1993	Normile et al.	
5,226,176	A	7/1993	Westaway et al.	
5,227,893	A *	7/1993	Ett	358/400
5,231,492	A	7/1993	Dangi et al.	
5,237,460	A	8/1993	Miller et al.	
5,237,675	A	8/1993	Hannon, Jr.	
5,243,341	A	9/1993	Seroussi et al.	
5,243,348	A	9/1993	Jackson	
5,247,638	A	9/1993	O'Brien et al.	
5,247,646	A	9/1993	Osterlund et al.	
5,263,168	A	11/1993	Toms et al.	
5,270,832	A	12/1993	Balkanski et al.	
5,287,420	A	2/1994	Barrett	
5,293,379	A	3/1994	Carr	
5,307,497	A	4/1994	Feigenbaum et al.	
5,309,555	A	5/1994	Akins et al.	
5,355,498	A	10/1994	Provino et al.	
5,357,614	A	10/1994	Pattisam et al.	
5,379,036	A	1/1995	Storer	
5,379,757	A	1/1995	Hiyama et al.	
5,381,145	A	1/1995	Allen et al.	
5,394,534	A	2/1995	Kulakowski et al.	
5,396,228	A	3/1995	Garahi	
5,400,401	A *	3/1995	Wasilewski et al.	380/212
5,403,639	A *	4/1995	Belsan et al.	707/204
5,406,278	A	4/1995	Graybill et al.	
5,406,279	A	4/1995	Anderson et al.	
5,412,384	A	5/1995	Chang et al.	
5,414,850	A	5/1995	Whiting	
5,420,639	A	5/1995	Perkins	
5,434,983	A	7/1995	Yaso et al.	
5,452,287	A	9/1995	Dicecco	
5,461,679	A	10/1995	Normile et al.	
5,467,087	A	11/1995	Chu	
5,471,206	A	11/1995	Allen et al.	
5,479,587	A	12/1995	Campbell et al.	
5,483,470	A	1/1996	Alur et al.	
5,486,826	A	1/1996	Remillard	
5,495,244	A	2/1996	Je-chang et al.	
5,506,844	A	4/1996	Rao	
5,506,872	A	4/1996	Mohler	
5,530,845	A	6/1996	Hiatt	
5,533,051	A	7/1996	James	
5,535,356	A	7/1996	Kim et al.	
5,537,658	A	7/1996	Bakke et al.	
5,557,551	A	9/1996	Craft	
5,557,668	A	9/1996	Brady	
5,557,749	A	9/1996	Norris	
5,561,824	A	10/1996	Carreiro et al.	
5,563,961	A	10/1996	Rynderman et al.	
5,574,952	A	11/1996	Brady et al.	

5,574,953	A	11/1996	Rust et al.	
5,583,500	A	12/1996	Allen et al.	
5,590,306	A	12/1996	Watanabe et al.	
5,596,674	A	1/1997	Bhandari et al.	
5,604,824	A	2/1997	Chui et al.	
5,606,706	A	2/1997	Takamoto et al.	
5,611,024	A	3/1997	Campbell et al.	
5,612,788	A	3/1997	Stone	
5,613,069	A *	3/1997	Walker	709/238
5,615,017	A	3/1997	Choi	
5,621,820	A	4/1997	Rynderman et al.	
5,623,623	A	4/1997	Kim et al.	
5,623,701	A	4/1997	Bakke et al.	
5,627,534	A	5/1997	Craft	
5,627,995	A	5/1997	Miller et al.	
5,629,732	A	5/1997	Moskowitz et al.	
5,630,092	A	5/1997	Carreiro et al.	
5,635,632	A *	6/1997	Fay et al.	73/61.63
5,635,932	A *	6/1997	Shinagawa et al.	341/51
5,638,498	A	6/1997	Tyler et al.	
5,640,158	A	6/1997	Okayama et al.	
5,642,506	A	6/1997	Lee	
5,649,032	A	7/1997	Burt et al.	
5,652,795	A	7/1997	Dillon et al.	
5,652,857	A	7/1997	Shimoi et al.	
5,652,917	A	7/1997	Maupin et al.	
5,654,703	A	8/1997	Clark, II	
5,655,138	A	8/1997	Kikinis	
5,666,560	A	9/1997	Moertl et al.	
5,668,737	A	9/1997	Iler	
5,671,389	A	9/1997	Saliba	
5,675,333	A	10/1997	Boursier et al.	
5,686,916	A	11/1997	Bakhmutsky	
5,694,619	A	12/1997	Konno	
5,696,927	A	12/1997	MacDonald et al.	
5,703,793	A	12/1997	Wise et al.	
5,715,477	A	2/1998	Kikinis	
5,717,393	A	2/1998	Nakano et al.	
5,717,394	A	2/1998	Schwartz et al.	
5,719,862	A *	2/1998	Lee et al.	370/355
5,721,958	A	2/1998	Kikinis	
5,724,475	A	3/1998	Kirsten	
5,729,228	A	3/1998	Franaszek et al.	
5,748,904	A	5/1998	Huang et al.	
5,757,852	A	5/1998	Kericevic et al.	
5,771,340	A	6/1998	Nakazato et al.	
5,778,411	A	7/1998	DeMoss et al.	
5,781,767	A	7/1998	Inoue et al.	
5,784,572	A	7/1998	Rostoker et al.	
5,787,487	A	7/1998	Hashimoto et al.	
5,796,864	A *	8/1998	Callahan	382/166
5,799,110	A	8/1998	Israelsen et al.	
5,805,932	A	9/1998	Kawashima et al.	
5,808,660	A	9/1998	Sekine et al.	
5,809,176	A	9/1998	Yajima	
5,809,337	A	9/1998	Hannah et al.	
5,812,789	A	9/1998	Diaz	
5,818,368	A	10/1998	Langley	
5,818,369	A	10/1998	Withers	
5,818,530	A	10/1998	Canfield et al.	
5,819,215	A	10/1998	Dobson et al.	
5,825,424	A	10/1998	Canfield et al.	
5,825,830	A	10/1998	Kopf	375/340
5,832,037	A	11/1998	Park	
5,832,126	A	11/1998	Tanaka	
5,836,003	A	11/1998	Sadeh	
5,838,996	A	11/1998	deCarmo	
5,839,100	A	11/1998	Wegener	
5,841,979	A	11/1998	Schulhof et al.	
5,847,762	A	12/1998	Canfield et al.	
5,861,824	A	1/1999	Ryu et al.	
5,861,920	A	1/1999	Mead et al.	
5,864,342	A	1/1999	Kajiya et al.	

5,867,167 A * 2/1999 Deering 345/419
5,867,602 A 2/1999 Zandi et al.
5,870,036 A 2/1999 Franaszek et al.
5,870,087 A 2/1999 Chau
5,872,530 A 2/1999 Domyo et al.
5,883,975 A 3/1999 Narita et al.
5,886,655 A 3/1999 Rust
5,889,961 A 3/1999 Dobbek
5,915,079 A 6/1999 Vondran, Jr. et al.
5,917,438 A 6/1999 Ando
5,920,326 A 7/1999 Rentschler et al.
5,936,616 A 8/1999 Torborg, Jr. et al.
5,949,355 A 9/1999 Panaoussis
5,955,976 A 9/1999 Heath
5,960,465 A 9/1999 Adams
5,964,842 A 10/1999 Packard
5,968,149 A 10/1999 Jaquette et al.
5,973,630 A 10/1999 Heath
5,974,235 A 10/1999 Nunally et al.
5,974,471 A 10/1999 Belt
5,978,483 A 11/1999 Thompson, Jr. et al.
5,982,723 A 11/1999 Kamatani
5,991,515 A 11/1999 Fall et al.
5,996,033 A 11/1999 Chiu-Hao
6,000,009 A 12/1999 Brady
6,002,411 A 12/1999 Dye
6,003,115 A 12/1999 Spear et al.
6,008,743 A 12/1999 Jaquette
6,011,901 A 1/2000 Kirsten
6,014,694 A 1/2000 Aharoni et al.
6,026,217 A 2/2000 Adiletta
6,028,725 A 2/2000 Blumenau
6,031,939 A 2/2000 Gilbert et al.
6,032,148 A 2/2000 Wilkes
6,061,398 A 5/2000 Satoh et al.
6,073,232 A 6/2000 Kroeker et al.
6,075,470 A 6/2000 Little et al.
6,091,777 A 7/2000 Guetz et al.
6,094,634 A 7/2000 Yahagi et al.
6,097,520 A 8/2000 Kadnier
6,104,389 A 8/2000 Ando
6,105,130 A 8/2000 Wu et al.
6,128,412 A 10/2000 Satoh
6,141,053 A 10/2000 Saukkonen
6,145,069 A 11/2000 Dye
6,169,241 B1 1/2001 Shimizu
6,172,936 B1 1/2001 Kitazaki
6,173,381 B1 1/2001 Dye
6,182,125 B1 * 1/2001 Borella et al. 709/218
6,192,082 B1 2/2001 Moriarty et al.
6,195,024 B1 2/2001 Fallon
6,195,465 B1 2/2001 Zandi et al.
6,222,886 B1 4/2001 Yogeshwar
6,225,922 B1 5/2001 Norton 341/87
6,226,667 B1 5/2001 Matthews et al.
6,226,740 B1 5/2001 Iga
6,253,264 B1 6/2001 Sebastian
6,272,178 B1 8/2001 Nieweglowski et al.
6,272,627 B1 8/2001 Mann
6,272,628 B1 8/2001 Aguilar et al.
6,282,641 B1 8/2001 Christensen
6,308,311 B1 10/2001 Carmichael et al.
6,309,424 B1 10/2001 Fallon
6,317,714 B1 11/2001 Del Castillo et al.
6,330,622 B1 12/2001 Shaefer
6,345,307 B1 2/2002 Booth
6,392,567 B2 5/2002 Satoh
6,404,931 B1 6/2002 Chen et al.
6,421,387 B1 7/2002 Rhee
6,434,168 B1 8/2002 Kari
6,434,695 B1 8/2002 Esfahani et al.
6,442,659 B1 8/2002 Blumenau
6,449,682 B1 9/2002 Toorians
6,452,602 B1 9/2002 Morein
6,463,509 B1 10/2002 Teoman et al.
6,487,640 B1 11/2002 Lipasti
6,489,902 B2 12/2002 Heath
6,513,113 B1 1/2003 Kobayashi
6,529,633 B1 3/2003 Easwar et al.
6,532,121 B1 3/2003 Rust et al.
6,539,456 B2 3/2003 Stewart
6,542,644 B1 4/2003 Satoh
6,577,254 B2 6/2003 Rasmussen 341/51
6,590,609 B1 7/2003 Kitade et al.
6,597,812 B1 7/2003 Fallon et al.
6,601,104 B1 7/2003 Fallon
6,604,040 B2 8/2003 Kawasaki et al.
6,604,158 B1 8/2003 Fallon
6,606,040 B2 8/2003 Abdat
6,606,413 B1 8/2003 Zeineh
6,609,223 B1 * 8/2003 Wolfgang 714/752
6,618,728 B1 9/2003 Rail
6,624,761 B2 9/2003 Fallon
6,650,261 B2 11/2003 Nelson et al.
6,661,839 B1 12/2003 Ishida et al.
6,661,845 B1 12/2003 Herath
6,704,840 B2 3/2004 Nalawadi et al.
6,711,709 B1 3/2004 York
6,717,534 B2 4/2004 Yokose
6,731,814 B2 5/2004 Zeck et al.
6,745,282 B2 6/2004 Okada
6,748,457 B2 6/2004 Fallon
6,756,922 B2 6/2004 Ossia
6,810,434 B2 10/2004 Muthujumaraswathy et al.
6,885,316 B2 4/2005 Mehring
6,885,319 B2 4/2005 Geiger et al.
6,909,383 B2 6/2005 Shokrollahi et al.
6,944,740 B2 9/2005 Abali et al.
7,102,544 B1 9/2006 Liu
7,130,913 B2 10/2006 Fallon
7,161,506 B2 1/2007 Fallon
7,181,608 B2 2/2007 Fallon et al.
7,190,284 B1 3/2007 Dye et al.
7,321,937 B2 1/2008 Fallon
2001/0031092 A1 10/2001 Zeck et al.
2001/0032128 A1 10/2001 Kepecs
2001/0052038 A1 12/2001 Fallon et al.
2002/0037035 A1 3/2002 Singh
2002/0080871 A1 6/2002 Fallon et al.
2002/0101367 A1 8/2002 Geiger et al.
2002/0104891 A1 8/2002 Otto
2002/0126755 A1 9/2002 Li et al.
2002/0191692 A1 12/2002 Fallon et al.
2003/0030575 A1 2/2003 Frachtenberg et al.
2003/0034905 A1 2/2003 Anton
2003/0084238 A1 5/2003 Okada et al.
2003/0142874 A1 7/2003 Schwartz
2003/0191876 A1 10/2003 Fallon
2004/0073710 A1 4/2004 Fallon
2006/0015650 A1 1/2006 Fallon
2006/0181441 A1 8/2006 Fallon
2006/0181442 A1 8/2006 Fallon
2006/0184696 A1 8/2006 Fallon
2006/0190644 A1 8/2006 Fallon
2006/0195601 A1 8/2006 Fallon
2007/0043939 A1 2/2007 Fallon et al.
2007/0050514 A1 3/2007 Fallon
2007/0050515 A1 3/2007 Fallon
2007/0067483 A1 3/2007 Fallon
2007/0083746 A1 4/2007 Fallon et al.
2007/0109154 A1 5/2007 Fallon
2007/0109155 A1 5/2007 Fallon
2007/0109156 A1 5/2007 Fallon

2007/0174209 A1 7/2007 Fallon et al.

FOREIGN PATENT DOCUMENTS

EP	0164677		12/1985
EP	0185098		6/1986
EP	0283798		9/1988
EP	0405572	A2	1/1991
EP	0405572	A3	3/1991
EP	0493130		7/1992
EP	0587437		3/1994
EP	0595406		5/1994
EP	0718751	A2	6/1996
EP	0718751	A3	2/1997
GB	2162025		1/1986
JP	6051989		2/1994
JP	9188009		7/1997
JP	11149376		6/1999
WO	WO9414273		6/1994
WO	WO9429852		12/1994
WO	WO9502873		1/1995
WO	WO9748212		12/1997

OTHER PUBLICATIONS

Pen-Shu Yeh, *The CCSDS Lossless Data Compression Recommendation for Space Applications*, Chapter 16, Lossless Compression Handbook, Elsevier Science (USA), 2003, pp. 311-326, No Month.

Robert F. Rice, *Some Practical Universal Noiseless Coding Techniques*, Jet Propulsion Laboratory, Pasadena, California, JPL Publication 79-22, Mar. 15, 1979.

K. Murashita et al., High-Speed Statistical Compression using Self-organized Rules and Predetermined Code Tables, IEEE, 1996 Data Compresson conference, No Month.

IBM, Fast Dos Soft Boot, Feb. 1, 1994, vol. 37, Issue 28, pp. 185-186.

J. Anderson et al. Codec squeezes color teleconferencing through digital telephone lines, Electronics 1984, pp. 13-15, No Month.

Robert Rice, “Lossless Coding Standards For Space Data Systems”, IEEE 1058-6393/97, pp. 577-585, No Date.

Coene, W et al. “A Fast Route For Application of Rate-distortion Optimal Quantization in an MPEG Video Encoder” Proceedings of the International Conference on Image Processing, US., New York, IEEE, Sep. 16, 1996, pp. 825-828.

“Operating System Platform Abstraction Method”, IBM Technical Disclosure Bulletin, Feb. 1995, vol. 38, Issue No. 2, pp. 343-344.

Millman, Howard, “Image and video compression”, Computerworld, vol. 33, Issue No. 3, Jan. 18, 1999, pp. 78.

“IBM boosts your memory”, Geek.com [online], Jun. 26, 2000 [retrieved on Jul. 6, 2007], <URL: http://www.geek.com/ibm-boosts-your-memory/>.

“IBM Research Breakthrough Doubles Computer Memory Capacity”, IBM Press Release [online], Jun. 26, 2000 [retrieved on Jul. 6, 2007], <URL: http://www-03.ibm.com/press/us/en/pressrelease/1653.wss>.

“ServerWorks To Deliver IBM’s Memory eXpansion Technology in Next-Generation Core Logic for Servers”, ServerWorks Press Release [online], Jun. 27, 2000 [retrieved on Jul. 14, 2000], <URL: http://www.serverworks.com/news/press/ 000627.html>.

Abali, B., et al., “Memory Expansion Technology (MXT) Software support and performance”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 287-301.

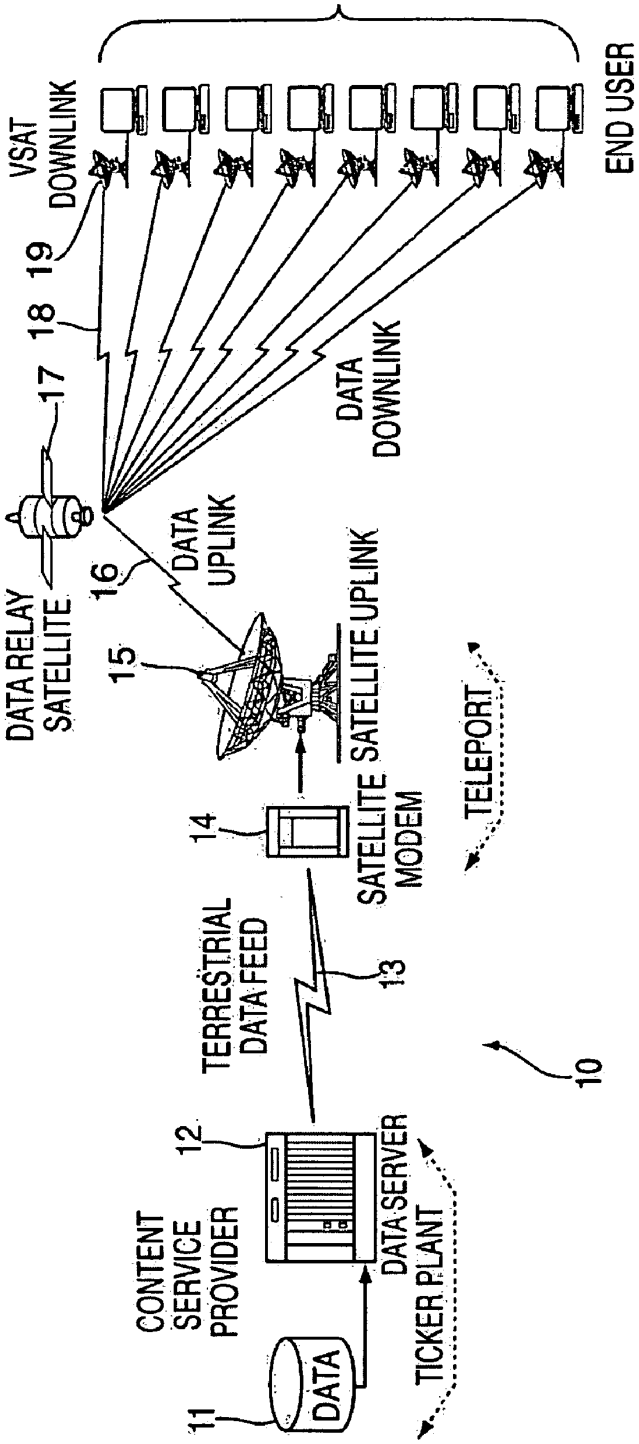
Franaszek, P.A., et al., “Algorithms and data structures for compressed-memory machines”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 245-258.

Franaszek, P.A., et al., “On internal organization in compressed random-access memories”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 259-270.

Smith, T.B., et al., “Memory Expansion Technology (MXT) Competitive impact”, IBM Journal of Research and Development, Vo. 45, Issue No. 2, Mar. 2001, pp. 303-309.

Tremaine, R. B., et al., “IBM Memory Expansion Technology (MXT)”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 271-285.

* cited by examiner



VSAT DOWNLINK
END USER
19
18
17
DATA DOWNLINK
DATA RELAY SATELLITE
16
DATA UPLINK
15
SATELLITE UPLINK
TELEPORT
14
SATELLITE MODEM
TERRESTRIAL DATA FEED
13
10
12
CONTENT SERVICE PROVIDER
DATA SERVER
TICKER PLANT
11
DATA

FIG. 1

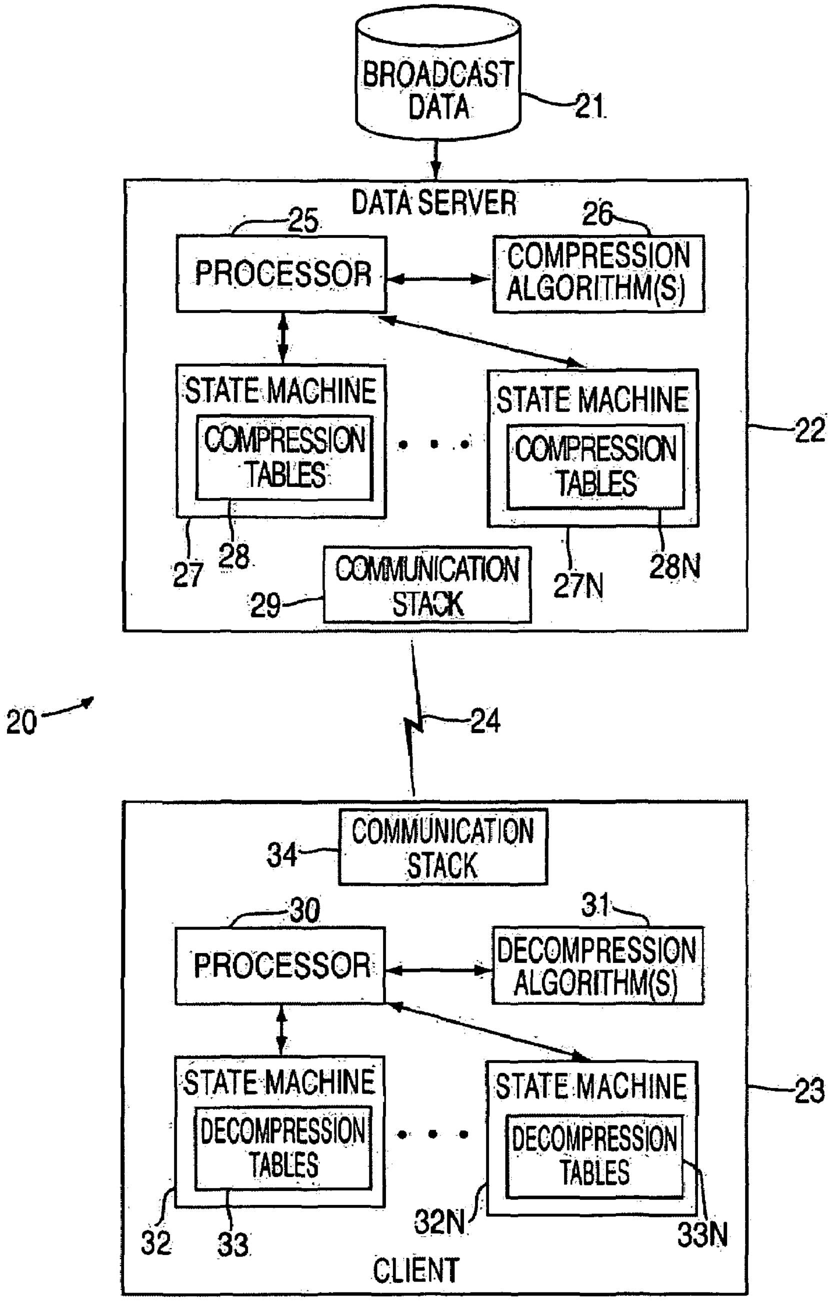

BROADCAST DATA
21
DATA SERVER
25
26
PROCESSOR
COMPRESSION ALGORITHM(S)
STATE MACHINE
COMPRESSION TABLES
STATE MACHINE
COMPRESSION TABLES
22
27
28
COMMUNICATION STACK
29
27N
28N
20
24
COMMUNICATION STACK
34
30
31
PROCESSOR
DECOMPRESSION ALGORITHM(S)
STATE MACHINE
DECOMPRESSION TABLES
STATE MACHINE
DECOMPRESSION TABLES
23
32
33
32N
33N
CLIENT

FIG. 2

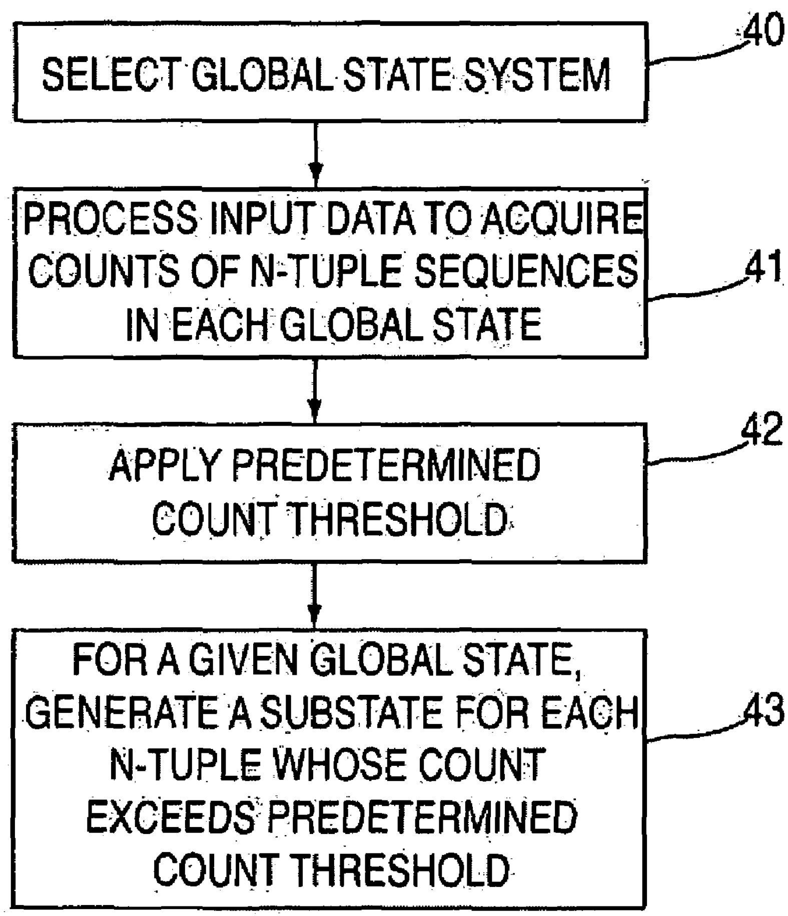
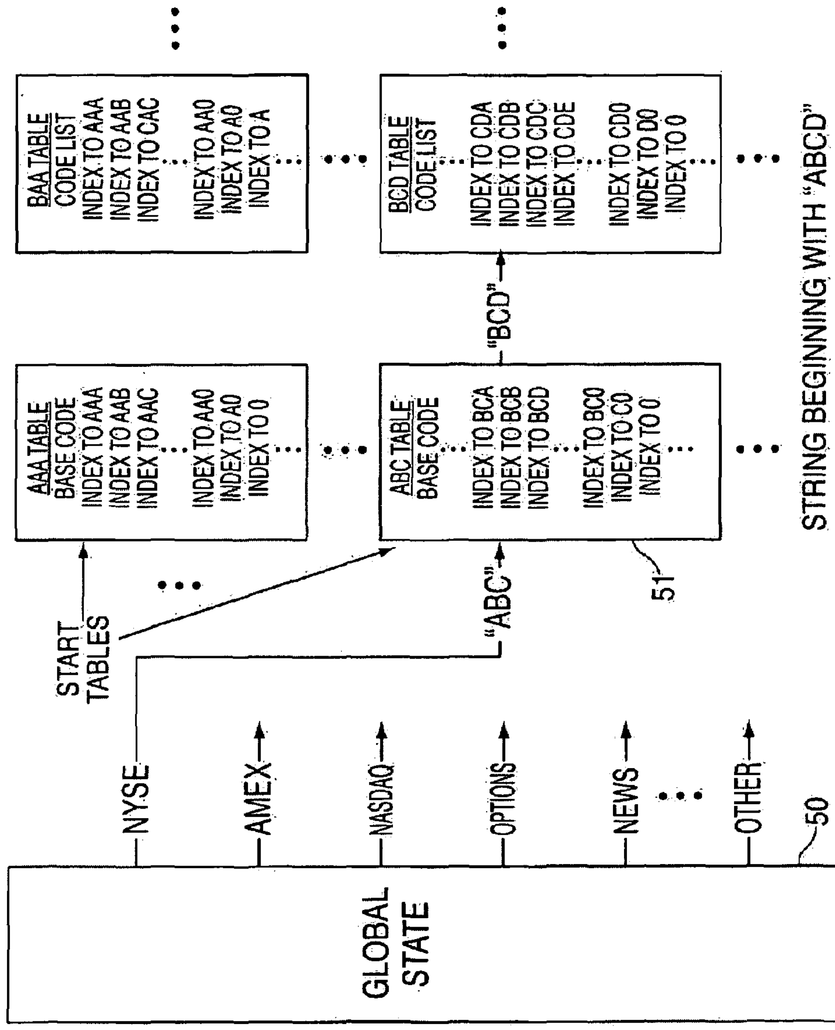

SELECT GLOBAL STATE SYSTEM
40
PROCESS INPUT DATA TO ACQUIRE COUNTS OF N-TUPLE SEQUENCES IN EACH GLOBAL STATE
41
APPLY PREDETERMINED COUNT THRESHOLD
42
FOR A GIVEN GLOBAL STATE, GENERATE A SUBSTATE FOR EACH N-TUPLE WHOSE COUNT EXCEEDS PREDETERMINED COUNT THRESHOLD
43

FIG. 3



GLOBAL STATE
50
NYSE
AMEX
NASDAQ
OPTIONS
NEWS
OTHER
START TABLES
"ABC"
51
AAA TABLE
BASE CODE
INDEX TO AAA
INDEX TO AAB
INDEX TO AAC
INDEX TO AA0
INDEX TO A0
INDEX TO 0
ABC TABLE
BASE CODE
INDEX TO BCA
INDEX TO BCB
INDEX TO BCD
INDEX TO BC0
INDEX TO C0
INDEX TO 0
"BCD"
BAA TABLE
CODE LIST
INDEX TO AAA
INDEX TO AAB
INDEX TO CAC
INDEX TO AA0
INDEX TO A0
INDEX TO A
BCD TABLE
CODE LIST
INDEX TO CDA
INDEX TO CDB
INDEX TO CDC
INDEX TO CDE
INDEX TO CD0
INDEX TO D0
INDEX TO 0
STRING BEGINNING WITH "ABCD"

FIG. 4

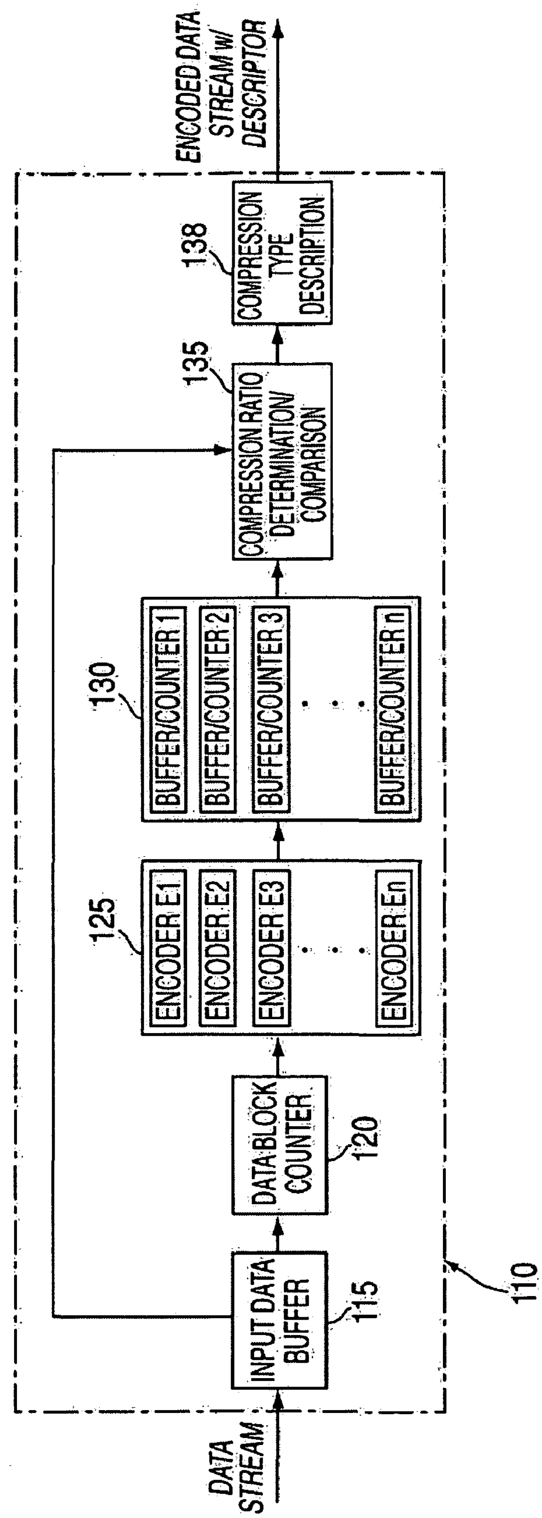

DATA STREAM
INPUT DATA BUFFER
115
DATA BLOCK COUNTER
120
125
ENCODER E1
ENCODER E2
ENCODER E3
ENCODER En
130
BUFFER/COUNTER 1
BUFFER/COUNTER 2
BUFFER/COUNTER 3
BUFFER/COUNTER n
135
COMPRESSION RATIO DETERMINATION/ COMPARISON
138
COMPRESSION TYPE DESCRIPTION
ENCODED DATA STREAM w/ DESCRIPTOR
110

FIG. 5

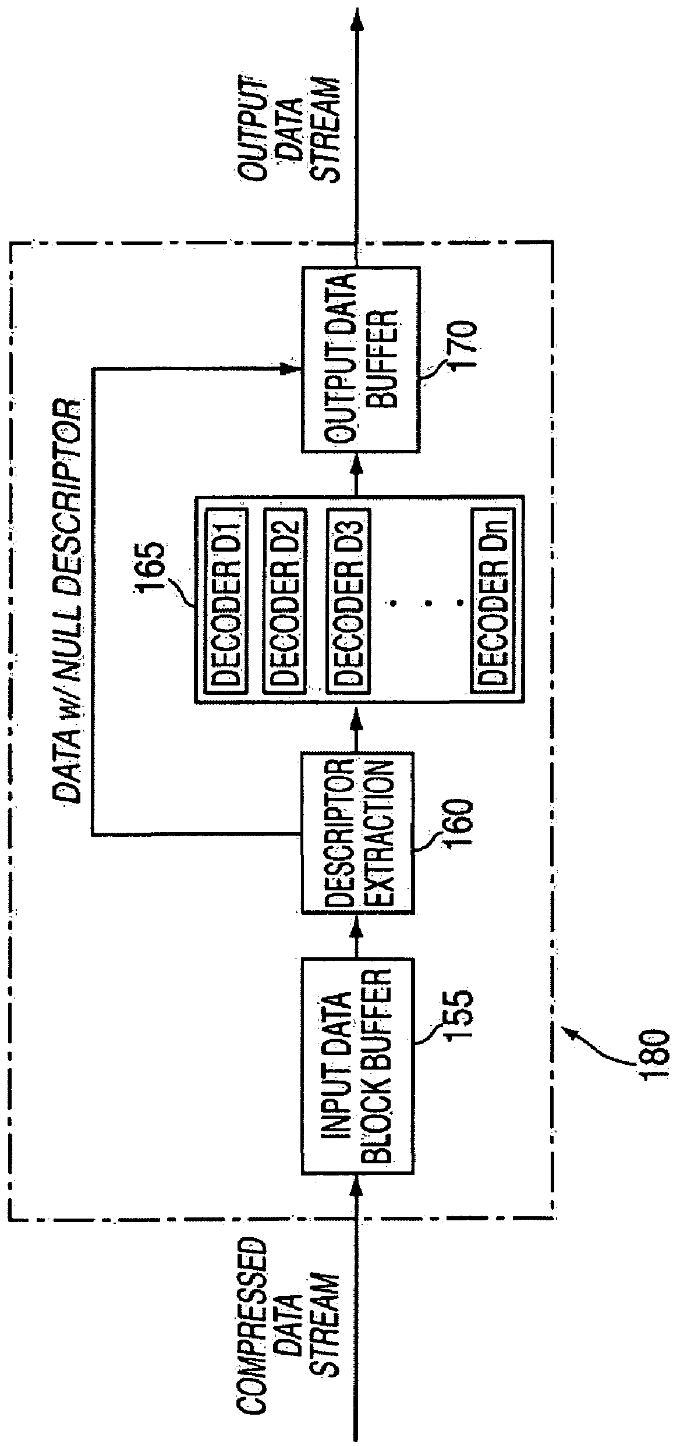

COMPRESSED DATA STREAM
INPUT DATA BLOCK BUFFER
155
DESCRIPTOR EXTRACTION
160
DATA w/ NULL DESCRIPTOR
165
DECODER D1
DECODER D2
DECODER D3
DECODER Dn
OUTPUT DATA BUFFER
170
OUTPUT DATA STREAM
180

FIG. 6

SYSTEM AND METHOD FOR DATA FEED ACCELERATION AND ENCRYPTION

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Continuation-in-Part of U.S. patent application Ser. No. 09/969,987, filed on Oct. 3, 2001, which claims the benefit of U.S. Provisional Application No. 60/237,571, filed on Oct. 3, 2000, both of which are fully incorporated herein by reference. In addition, this application claims the benefit of U.S. Provisional Application No. 60/378,517, filed on May 7, 2002, which is fully incorporated herein by reference.

TECHNICAL FIELD

The present invention relates generally to systems and method for providing data transmission, and in particular, to systems and method for providing accelerated transmission of data, such as financial trading data, financial services data, financial analytical data, company background data and news feeds, advertisements, and all other forms or information over a communication channel using data compression and decompression to provide data broadcast feeds, bi-directional data transfers, and all other forms of communication with or without security and effectively increase the bandwidth of the communication channel and/or reduce the latency of data transmission.

BACKGROUND

The financial markets and financial information services industry encompass a broad range of financial information ranging from basic stock quotations, bids, order, fulfillment, financial and quotations to analyst reports to detailed pricing of Treasury Bills and Callable bonds. Users of financial information can now generally be divided into three segments—Traders, Information Users and Analytics Users, although some users constitute components from one or more of these categories.

Traders utilize data from financial markets such as NASDAQ, the American Stock Exchange, the New York Stock Exchange, the Tokyo Exchange, the London Exchange, the Chicago Options Board, and similar institutions that offer the ability to buy and sell stocks, options, futures, bonds, derivatives, and other financial instruments. The need for vast quantities of information is vital for making informed decisions and executing optimal transactions.

Thus given the importance of receiving this information over computer networks, an improved system and method for providing secure point-to-point solution for transparent multiplication of bandwidth over conventional communication channels is highly desirable.

For example, with the introduction of Nasdaq's next generation trading system SuperMontage, Nasdaq will offer market data users an unparalleled view into the activity, liquidity, and transparency of the Nasdaq market.

For example, currently Nasdaq provides each market participant's best-attributed quotation in each stock in which it makes a market. This system known as SuperMontage allows Nasdaq to accept multiple orders from each market participate in each stock for execution within SuperMontage. Nasdaq offers that data, with multiple levels of interest from individual market participants, through new data services.

Nasdaq provides this data on both an aggregated and a detailed basis for the top five price levels in SuperMontage. This data is currently offered through market data vendors and broker/dealer distributors via the following four entitlement packages:

QuoteView SM	Each SuperMontage participant's best bid and offer, as well as the best bid and offer available on SuperMontage.
DepthView SM	The aggregate size, by price level, of all Nasdaq market participants' attributed and unattributed quotations/orders that are in the top five price levels in SuperMontage.
PowerView SM	Bundled QuoteView and DepthView.
TotalView SM	PowerView plus all Nasdaq market participants' attributed quotations/orders that are in the top five price levels in SuperMontage, in addition to the aggregate size of all unattributed quotes/orders at each of the top five price levels.

The NASDAQ SuperMontage trading system has been cited to be representative of trend for explosive growth in the quantity of information for all emergent and future trading and financial information distribution systems. Increases in processing power at the end user sites will allow traders, analysts, and all other interested parties to process substantially larger quantities of data in far shorter periods of time, increasing the demand substantially.

The ever increasing need for liquidity in the financials markets, coupled with the competitive pressures on reducing bid/ask spreads and instantaneous order matching/fulfillment, along the need for synchronized low latency data dissemination makes the need for the present invention ever more important. Depth of market information, required to achieve many of these goals requires orders of magnitude increases in Realtime trade information and bid/ask pricing (Best, 2^{nd} best, . . .).

A fundamental problem within the current art is the high cost of implementing, disseminating, and operating trading systems such as SuperMontage within the financial services industry. This is in large part due to the high bandwidth required to transfer the large quantities of data inherent in the operation of these systems. In addition the processing power required to store, transmit, route, and display the information further compounds cost and complexity.

This fundamental problem is in large part the result of utilizing multiple simultaneous T1 lines to transmit data. The data must be multiplexed into separate data streams, transmitted on separate data lines, and de-multiplexed and checked. Software solutions have high latency and cost while hardware solutions have even higher cost and complexity with somewhat lower latency. In addition the synchronization and data integrity checking require substantial cost, complexity, inherent unreliability, and latency. These and other limitations are solved by the present invention.

Further compounding this issue is a globalization and consolidation taking place amongst the various financial exchanges. The emergence of localized exchanges (ECNS—Electronic Computer Networks) coupled with the goal of 24 hour/7 day global trading will, in and of itself, drive another exponential increase in long haul international bandwidth requirements, while ECNs and other localized trading networks will similarly drive domestic bandwidth requirements. Clearly long haul links are orders of magnitude more expensive than domestic links and the value and significance of the present invention is at least proportionately more important.

Information users range from non-finance business professionals to curious stock market investors and tend to seek basic financial information and data. Analytical users on the other hand, tend to be finance professionals who require more

arcane financial information and utilize sophisticated analytical tools to manipulate and analyze data (e.g. for writing option contracts).

Historically, proprietary systems, such as Thomson, Bloomberg, Reuters and Bridge Information, have been the primary electronic source for financial information to both the informational and analytical users. These closed systems required dedicated telecommunications lines and often product-specific hardware and software. The most typical installations are land-based networking solutions such as T1, or ISDN, and satellite-based "wireless" solutions at speeds of 384 kbps.

Latency of financial data is critical to the execution of financial transactions. Indeed the more timely receipt of financial data from various sources including the New York Stock Exchange, American Stock Exchange, National Association of Securities Dealers (NASDAQ), Options Exchange, Commodities Exchanges, and Futures presents a fundamental advantage to those who trade. Latency is induced by the long time taken transmit and receive uncompressed data or to compress and encrypt data prior to transmission, along with the associated time to decrypt and decompress. Often current methods of encryption and compression take as much or substantially more time than the actual time to transmit the uncompressed, unencrypted data. Thus another problem within the current art is the latency induced by the act of encryption, compression, decryption, and decompression. The present invention overcomes this limitation within the current art.

Modern data compression algorithms suffer from poor compression, high latency, or both. Within the present art algorithms such as Lempel-Ziv, modified/embellished Lempel-Ziv, Binary Arithmetic, and Huffman coding are essentially generic algorithm having a varied effectiveness on different data types. Also small increases in compression to the negentropy limit of the data generally require exponentially greater periods of time and substantially higher latency. Negentropy is herein defined as the information content within a given piece of data. Generic algorithms are currently utilized as data types and content format is constantly changed within the financial industry. Many changes are gradual however there are also abrupt changes, such as the recent switch to decimalization to reduce granularity that has imposed substantial requirements on data transmission bandwidth infrastructure within the financial industry. Thus another problem within the current art is the high latency and poor compression due to the use of generic data compression algorithms on financial data and news feeds. This limitation is also overcome by the present invention.

Within the financial and news feeds, data is often segregated into packets for transmission. Further, in inquiry-response type systems, as found in many financial research systems, the size of request packets and also response packets is quite small. As such, response servers often wait for long periods of time (for example 500 msec) to aggregate data packets prior to transmission back to the inquirer. By aggregating the data, and then applying compression, somewhat higher compression ratios are often achieved. This then translates to lower data communications costs or more customers served for a given amount of available communications bandwidth. Thus another problem within the current art is the substantial latency caused by aggregating data packets due to poor data compression efficiency and packet overhead. This limitation is also solved by the present invention.

Another problem within the current art is the need for data redundancy. Currently many trading systems utilize two independent links to compare data to verify integrity. Second, the bandwidth of discrete last mile links, typically T1s, is limited to 1.5 Megabits/second.

Increases in bandwidth beyond this point require complex protocols to fuse data from multiple links, adding cost and complexity, while also increasing latency and inherent data error rates. This limitation is also solved by the present invention.

Another limitation within the current art is that nearly all financial institutions use one or more T1 lines to transfer information to and from their customers. While the costs of bandwidth have moderately decreased over recent years this trend is slowing and the need forever increased bandwidth will substantively overshadow any future reductions. Indeed with the recent fall-out of the telecommunications companies the data communications price wars will end and we could easily see an increase in the cost of bandwidth. US Domestic T1 lines currently range from several hundred dollars to upwards of a thousand dollars per link, dependent upon quantity of T1 lines purchased, geographic location, length of connection, and quality/conditioning of line. Fractional T1 lines may also be purchased in 64 Kilobit/second increments with some cost savings.

A standard T1 line transmits data at a rate of 1.544 megabits per second. Accounting for framing and data transmission overhead this means that a T1 line is capable of transmitting a 150 Kilobytes per second. While 30× faster than a modem line (which provides only 5 kilobytes per second), both are relatively slow in relation to any reasonable level of information flow. For example, transferring the contents of data on a single CDROM would take well over an hour!

Thus it is likely that the capacity of many existing T1 lines will be exceeded in the near future. For our current example let's assume that we need to double the capacity of a T1 line. Normally this is done by adding a second T1 line and combining the contents of both with Multi-Link Point to Point Protocol (MLPP) or another relatively complex protocol. Within the current art this is neither necessary nor desirable. In fact any increase over the current limitation of a T1 line results in the addition of a second line. This limitation is overcome by the present invention.

Another limitation with the current art is the extraordinary bandwidth required for real-time (hot) co-location processing which has been dramatically increased as a result of the acts of terror committed against the United States of America on Sep. 11, 2001. In order for the redundancy of any co-location to be effective, it must be resident in a geographically disparate location; this could be a different state, a different coast, or even a different country. The trend towards globalization will further compound the need for the ability to simultaneously process transactions at geographically diverse co-locations.

It is a widely known fact within the financial industry that the overall throughput of transactions is governed by the bandwidth and latency of the co-location data link, along with delays associated with synchronization, i.e. the transaction must be complete at both locations and each location must know that the other location is complete before the transaction is finalized.

High bandwidth links such as T3's are often utilized as part of this backbone structure. A single T3 line has the bandwidth of Twenty-Eight T1 lines (28×1.544=43.232 megabits/second). Thus, in the best case, a T3 line is capable of transmitting 5.4 megabytes/second. By way of comparison, the contents of a single CDROM may be transferred in approximately two minutes with a T3 link. As stated earlier, a single T1 line would take over an hour to transmit the same quantity of data.

The volume of real-time data that is required to operate any major financial institution is staggering by comparison. To deal with this issue only critical account and transaction information is currently processed by co-locations in real-time. In fact, many institutions use batch mode processing where the transactions are only repeated "backed up" at the co-locations some time period later, up to 15 minutes or longer. The limitation of highly significant bandwidth and/or long delays with co-location processing and long latency times is solved by the present invention.

Thus given the importance of receiving financial information over computer networks, an improved system and method for providing secure point-to-point solution for transparent multiplication of bandwidth over conventional communication channels is highly desirable.

As previously stated, these and other limitations within the current art are solved by the present invention.

SUMMARY OF THE INVENTION

The present invention is directed to systems and methods for providing accelerated data transmission, and in particular to systems and methods of providing accelerated transmission of data, such as financial trading data, financial services data, financial analytical data, company background data, news, advertisements, and all other forms of information over a communications channel utilizing data compression and decompression to provide data transfer (secure or non-secure) and effectively increase the bandwidth of the communication channel and/or reduce the latency of data transmission. The present invention is universally applicable to all forms of data communication including broadcast type systems and bi-directional systems of any manner and any number of users or sites.

These and other aspects, features and advantages, of the present invention will become apparent from the following detailed description of preferred embodiments that is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF PREFERRED EMBODIMENTS

FIG. **1** is a block diagram of a system in which the present invention may be implemented for transmitting broadcast data;

FIG. **2** is a block diagram of a system and method for providing accelerated transmission of data over a communication channel according to an embodiment of the present invention;

FIG. **3** is a flow diagram illustrating a method for generating compression/decompression state machines according to one aspect of the present invention;

FIG. **4** is a diagram illustrating an exemplary encoding table structure according to the present invention, which may be generated using the process of FIG. **3**.

FIG. **5** is a diagram of a system/method for providing content independent data compression, which may be implemented for providing accelerated data transmission according to the present invention; and

FIG. **6** is a diagram of a system/method for providing content independent data decompression, which may be implemented for providing accelerated data transmission according to the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention is directed to systems and methods for providing accelerated transmission of broadcast data, such as financial data and news feeds, over a communication channel using data compression and decompression to provide secure transmission and transparent multiplication of communication bandwidth, as well as reduce the latency associated with data transmission of conventional systems.

In this disclosure, the following patents and patent applications, all of which are commonly owned, are fully incorporated herein by reference: U.S. Pat. Nos. 6,195,024, issued on Feb. 27, 2001, and 6,309,424, issued on Oct. 30, 2001 and U.S. patent application Ser. Nos. 10/076,013 filed on Feb. 13, 2002, 10/016,355, filed on Oct. 29, 2001, 09/481,243 filed on Jan. 11, 2000, and 09/266,394 filed on Mar. 11, 1999.

In general, the term "accelerated" data transmission refers to a process of receiving a data stream for transmission over a communication channel, compressing the broadcast data stream in real-time (wherein the term "real time" as used herein collectively refers to substantially real time, or at real time, or greater than real time) at a compression rate that increases the effective bandwidth of the communication channel, and transmitting the compressed broadcast data over the communication channel. The effective increase in bandwidth and reduction of latency of the communication channel is achieved by virtue of the fast than real-time, real-time, near real time, compression of a received data stream prior to transmission.

For instance, assume that the communication channel has a bandwidth of "B" megabytes per second. If a data transmission controller is capable of compressing (in substantially real time, real time, or faster than real time) an input data stream with an average compression rate of 3:1, then data can be transmitted over the communication channel at an effective rate of up to 3*B megabytes per second, thereby effectively increasing the bandwidth of the communication channel by a factor of three.

Further, when the receiver is capable decompressing (in substantially real time, real time, or faster than real time) the compressed data stream at a rate approximately equal to the compression rate, the point-to-point transmission rate between the transmitter and receiver is transparently increased. Advantageously, accelerated data transmission can mitigate the traditional bottleneck associated with, e.g., local and network data transmission.

If the compression and decompression are accomplished in real-time or faster, the compressed, transmitted and decompressed data is available before the receipt of an equivalent uncompressed stream. The "acceleration" of data transmission over the communication channel is achieved when the total time for compression, transmission, and decompression, is less than the total time for transmitting the data in uncompressed form. The fundamental operating principle of data acceleration is governed by the following relationship:

$$[T_{Compress}+T_{Transmit\ Accelerated}+T_{Decompress}]<T_{Transmit\ w/o\ Compression} \quad \text{EQ [1]}$$

Where:

$T_{Compress}$ =	Time to Compress a Packet of Data
$T_{Transmit\ Accelerated}$ =	Time to Transmit Compressed Data Packet
$T_{Decompress}$ =	Time to Decompress the Compressed Data Packet

-continued

$T_{Transmit\ w/o\ Compression}$ =	Time to Transmit the Uncompressed (Original) Data Packet

As stated in Equation [1] above, if the time to compress, transmit, and decompress a data packet is less than the time to transmit the data in original format, then the delivery of the data is said to be accelerated.

In the above relationship, a fundamental premise is that all information is preferably fully preserved. As such, lossless data compression is preferably applied. While this disclosure is directed to transmission of data in financial networks, for example, the concept of "acceleration" may be applied to the storage and retrieval of data to any memory or storage device using the compression methods disclosed in the above-incorporated U.S. Pat. Nos. 6,195,024 and 6,309,424, and U.S. application Ser. No. 10/016,355, and the storage acceleration techniques disclosed in the above-incorporated application Ser. Nos. 09/481,243 and 09/266,394.

Returning to Equation [1], data acceleration depends on several factors including the creation of compression and decompression algorithms that are both effective (achieve good compression ratios) and efficient (operate rapidly with a minimum of computing processor and memory resources).

Rearranging the terms of Equation [1] we can see that the total time to transmit data in an "accelerated" form (transmit compressed data (is the sum of the original time to transmit the data in an uncompressed fashion divided by the actual compression ratio achieved, plus the time to compress and decompress the data.

$$T_{Transmit\ Accelerated}=[T_{ransmit\ w/o\ Compression}/CR]+T_{Compress}+T_{Decompress} \quad \text{EQ [2]}$$

Where:

CR=Compression Ratio

Thus the latency reduction is the simple arithmetic difference between the time to transmit the original data minus the total time to transmit the accelerated data (per Equation 2 above), resulting in:

$$T_{Latency\ Reduction}=T_{Transmit\ w/o\ Compression}-T_{Transmit\ Accelerated} \quad \text{EQ [3]}$$

And finally the achieved "Acceleration Ratio" is defined as:

$$\text{Acceleration Ratio}=T_{Transmit\ w/o\ Compression}/T_{Transmit\ Accelerated} \quad \text{EQ [4]}$$

A number of interesting observations come to light from these relatively simple algebraic relationships and are implemented within the present invention:

Compression Ratio: The present inventions achieve a consistent reduction in latency. The data compression ratio is substantial and repeatable on each data packet.

Compression Rate: The present invention achieves a consistent reduction in latency. Both the time to compress and decompress the data packet must be an absolute minimum, repeatable on each data packet, and always within predefined allowable bounds.

Packet Independence: The present invention has no packet-to-packet data dependency. By way of example, in UDP and Multicast operations there are no guarantees on delivery of data packets, nor on the order of delivered data packets. IP data packets, similarly, have no guarantee on the order of delivery also. Thus algorithms that rely on dictionaries (Zlib, Glib, Lempel Ziv, etc.) are inherently unreliable in any financial real-world financial data applications.

It is to be understood that the present invention may be implemented in various forms of hardware, software, firmware, or a combination thereof. Preferably, the present invention is implemented on a computer platform including hardware such as one or more central processing units (CPU) or digital signal processors (DSP), a random access memory (RAM), and input/output (I/O) interface(s). The computer platform may also include an operating system, microinstruction code, and dedicated processing hardware utilizing combinatorial logic or finite state machines. The various processes and functions described herein may be either part of the hardware, microinstruction code or application programs that are executed via the operating system, or any combination thereof.

It is to be further understood that, because some of the constituent system components described herein are preferably implemented as software modules, the actual system connections shown in the Figures may differ depending upon the manner in that the systems are programmed. General purpose computers, servers, workstations, personal digital assistants, special purpose microprocessors, dedicated hardware, or and combination thereof may be employed to implement the present invention. Given the teachings herein, one of ordinary skill in the related art will be able to contemplate these and similar implementations or configurations of the present invention.

It should be noted that the techniques, methods, and algorithms and teachings of the present invention are representative and the present invention may be applied to any financial network, trading system, data feed or other information system.

FIG. **1** is a diagram illustrating a system in which the present invention may be implemented. The system **10** comprises content **11** and data server **12** associated with a service provider of broadcast data. The content **11** comprises information that is processed by the data server **12** to generate a broadcast, e.g., a news feed or financial data feed. As explained in further detail below, the data server **12** employs data compression to encode/encrypt the broadcast data **11** prior to transmission over various communication channels to one or more client site systems **20** of subscribing users, which comprise the necessary software and hardware to decode/decrypt the compressed broadcast data in real-time. In the exemplary embodiment of FIG. **1**, the communication channels comprise a landline **13** that feeds the compressed broadcast data to a satellite system comprising modem **14** and an uplink system **15**, which provides a data uplink **16** to a relay **17**. The relay **17** provides data downlinks **18** to one or more downlink systems **19**.

Advantageously, the proprietary software used by the data server **12** to compress the data stream in real-time and software used by the workstations **19** to decompress the data stream in real-time effectively provides a seamless and transparent increase in the transmission bandwidth of the various communication channels used, without requiring modification of existing network infrastructure.

Referring now to FIG. **2**, a block diagram illustrates a system/method for providing accelerated transmission of data according to one embodiment of the present invention. More specifically, FIG. **2** illustrates embodiments of a broadcast data server (transmitter) and client system (receiver) for implementing accelerated transmission and real-time processing of broadcast data. Broadcast data **21** (comprising one or more different broadcast types) is processed by data server

22 prior to transmission to client **23** over a communication channel **24**. The data server **22** utilizes a processor **25** (e.g., microprocessor, digital signal processor, etc.) for executing one or more compression algorithms **26** for compressing (in real-time) the broadcast data **21** prior to transmission. In preferred embodiments, compression is achieved using Huffman or Arithmetic encoding, wherein one or more state machines **27**-**27***n* are constructed based on a-priori knowledge of the structure and content of one or more given broadcast and data feeds.

As explained in further detail below, each state machine **27**-**27***n* comprises a set of compression tables that comprise information for encoding the next character (text, integer, etc.) or sequence of characters in the broadcast data feed, as well as pointers which point to the next state (encoding table) based on the character or character sequence. As explained in greater detail below, a skeleton for each state machine **27**-**27***n* (nodes and pointers) is preferably built by finding sequences of characters (n-tuples) that frequently appear in a given data input. Once a skeleton has been determined, a large set of data is processed through the system and counts are kept of character n-tuples for each state. These counts are then used to construct the compression tables associated with the state machine to provide statistical compression. The compressed data is transmitted over the communication channel **24** via a communication stack using any suitable protocol (e.g., RTP (real time protocol) using RTCP (real-time control protocol), TCP/IP, UDP, or any real-time streaming protocol with suitable control mechanism).

Similarly, the client **23** comprises a processor **30** for executing one or more decompression algorithms **31**. Depending one the data feed type, one of a plurality of decompression state machines **32**-**32***n* are used to decompress the compressed data stream received by the client **23** via communication stack **34**. Each state machine **32**-**32***n* comprises a set of decompression tables **33**-**33***n* that comprise information for decode the next encoded character (or symbol) or sequence of symbols in the compressed broadcast data feed, as well as pointers which point to the next state based on the symbol or symbol sequence. For each compression state machine **27**-**27***n* in the data server, a corresponding decompression state machine **32**-**32***n* is needed in the client **23** to decompress the associated data stream.

Advantageously, a compression/decompression scheme according to the present invention using Huffman or Arithmetic encoding provides secure transmission via de facto or virtual "encryption" in a real-time environment. Indeed, virtual encryption is achieved by virtue of the fast, yet complex, data compression using Huffman tree, for example, without necessarily requiring actual encryption of the compressed data and decryption of the compressed data. Because of the time-sensitive nature of the market data, and the ever-changing and data-dependent nature of the arithmetic scheme, decryption is virtually impractical, or so complex and useless as to render the data worthless upon eventual decoding.

However, data compression using Huffman or Arithmetic encoding yields encoded data that is very difficult to decode than current encryption schemes such as plain text or simple bit shuffling codes as currently used by broadcast service providers. An attacker must have the compression model and the tables used to compress the data stream to be able to obtain useful information from it. Thus, at one level of security, the client-side decompression tables are preferably stored in encrypted form and are decrypted on being loaded into the processor **30** (e.g., general purpose processor, DSP, etc.) using an encryption/decryption key that is validated for a subscribing user. In this manner, a client will be unable to use the tables on other processors or sites or after terminating a service contract.

Since Huffman compression uses the same bit code for a character each time it appears in a given context, an attacker with a very large data set of compressed and uncompressed data could possibly reconstruct the tables, assuming the overall model were known. Arithmetic compression, on the other hand, generates different bit patterns for the same character in the same context depending on surrounding characters. Arithmetic encoding provides at least an order of magnitude more difficult to recover the tables from the compressed and uncompressed data streams.

The following is a detailed discussion of a compression scheme using Huffman or Arithmetic encoding for providing accelerated transmission of broadcast data according to one aspect of the present invention. It is to be appreciated that the present invention is applicable with any data stream whose statistical regularity may be captured and represented in a state machine model. For example, the present invention applies to packetized data streams, in which the packets are limited in type format and content.

In one embodiment using Huffman or Arithmetic encoding, each character or character sequence is encoded (converted to a binary code) based on the frequency of character or character sequence in a given "context". For a given context, frequently appearing characters are encoded with few bits while infrequently appearing characters are encoded with more bits. High compression ratios are obtained if the frequency distribution of characters in most contexts is highly skewed with few frequently appearing characters and many characters seldomly (or never) appear.

Referring now to FIG. **3**, a flow diagram illustrates a method for generating compression/decompression state machines according to one aspect of the present invention. The "context" in which a character (or character sequence) is encoded in a given broadcast stream is based on a "global state" that represents packet type and large-scale structure and the previous few characters. The first step in building a compression scheme involves selecting a global state system based on the packet structure of the broadcast model (step **40**). More specifically, a global state system is constructed based on a priori knowledge of the data stream model, e.g., the packet type frequency and structure of the broadcast model. By way of example, one model for financial data may comprise four global states representing: a beginning of packet, an options packet, a NYSE (New York Stock Exchange) packet and some other packet type. Further, additional codes may be added to the encoding tables to indicate global state transitions (e.g., for an end of packet code in the broadcast model). If there is internal structure to packets, such as a header with different statistics than the body, additional global states could be added.

Once a global state system is selected, training samples from an associated data stream are passed through the global model to acquire counts of frequencies of the occurrence of n-tuple character sequences ending in each of the model states (step **41**). In a preferred embodiment, the n-tuples comprise character sequences having 1, 2 and 3 characters. Using the acquired counts, sub-states (or "local states") of the predefined global states are constructed based on previous characters in the data stream. A local state may depend on either none, 1, 2, or 3 (or more) previous characters in the stream. To provide a practical limitation, a predetermined count threshold is preferably applied to the count data (step **42**) and only those sequences that occur more often than the count threshold are added as local states (step **43**). For example, if a

three-character sequence does not occur sufficiently frequently, the count for the last two characters is tested, etc.

It is to be understood that any character sequence length "n" may be implemented depending on the application. The longer the allowed character sequence, the more memory is needed to store the encoding tables and/or the lower the count threshold should be set.

As samples of the data are passed through the state model, character (and transition code) counts for each context are accumulated. These counts are used to build the Huffman or Arithmetic coding tables. The construction of the global and local models is an iterative process. The count threshold for forming local states can be adjusted depending on the application. For instance, a larger threshold will result in less local states but less compression as well. Further, a comparison of statistics in local or global states may suggest adding or deleting global states.

The construction of the global model requires knowledge of the data stream packet structure. The construction of the local states is automatic (once the threshold is set).

FIG. **4** is a diagram of an exemplary state diagram (or encoding table structure) according to the present invention, which may be generated using the process of FIG. **3**.

As noted above with reference to FIGS. **1** and **2**, a compression scheme according to the present invention may be implemented in any system to provide accelerated data transmission to multiple client site systems. Preferably, the client site systems may connect at any time, so minimal immediate history may be used (since a newly connected site must be able to pick up quickly). A system according to an embodiment of the present invention uses statistical compression (Huffman or Arithmetic coding) using fixed (or adaptive) tables based on the statistics of a data feed sample. As noted above, it has been determined that the statistical compression schemes described herein are well adapted for use with structured data streams having repetitive data content (e.g., stock symbols and quotes, etc.) to provide fast and efficient data compression/decompression.

The following discussion provides further details regarding the preparation of statistical-based encoding tables and their use for compression/decompression according to the present invention. During a data compression process, the selection of which encoding table to use for compression is preferably based on up to n (where n is preferably equal to 3) preceding characters of the message. In an exemplary broadcast model tested by the present inventors, a data stream comprises messages that begin with an ID code in the range 0-31 with the remainder of the message being characters in the range 32-127. It was found that approximately half of the messages in a given sample began with ID code 0x0c and half of the remainder began with ID code 0x0f. Thus, a separate encoding table is preferably used for a message ID code. Further, separate table sets are used for messages beginning with 0x0c and with 0x0f, with the remaining messages lumped together in another table.

Each table has an additional termination code. The termination code in a "start table" indicates the end of a compression block. The termination code in all other tables indicates the end of the message. Thus, the start table comprises 33 entries and all other tables have 97 entries.

Using one table for each 3-character context would require prohibitive amounts of memory. For example, a complete one-character context would require 33+3*97=324 tables. Then, a complete two-character context would require 324*97=31,428 tables. And finally, a complete three-character context would require 324*97*97=3,048,516 tables. Preferably, as described above, the application of a count threshold at each context size reduces the amount of tables. Only when a context occurs at greater than the threshold rate in the sample will a table be created for that context.

Each table entry includes a link to the next table to be used. For instance, in an "abc" context table, the entry for next character "d" would point to the "bed" table, if such table was created. If such table was not created, the entry for next character "d" would point to the "cd" table, if such table existed. If no "cd" table exists, the "d" table would be used and if that fails, a base table for the message type would be used.

For a client site system to pick up the broadcast feed at any time, clearly identifiable synchronization points are preferably included in the compressed data stream. In a preferred embodiment, data is compressed in blocks with each block comprising some number of complete messages. Preferably, each compressed block ends with at least four bytes with each bit being logic 1 and no interior point in the compressed block will comprise 32 consecutive 1 bits. The compressed block preferably begins with two bytes giving the decompressed size of the block shifted to guarantee that the first byte of the compressed block is not all 1's. Thus, to achieve synchronization, the client site system can scan the input compressed data stream for 4 bytes of 0xff, wherein the next byte not equal to 0xff is deemed the start of a compressed block. In other words, the receiver will accumulate the compressed data until at least a sequence of 4 bytes each having a value of 0xff is detected in the input stream, at which point decompression will commence on the compressed input stream.

In another embodiment of the present invention, if a compressed block is more than 6 bytes longer than the uncompressed data, the data block is transmitted uncompressed preceded by the shifted two-byte count with the high bit set and trailed by 4 bytes of 0xff.

The following is discussion of a method for preparing Huffman Tables according to one aspect of the present invention. The Huffman codes generated by a conventional optimal algorithm have been modified in various ways in accordance with the present invention. First, in order that there not be 32 consecutive one bits in the data stream except at the end of a compression block, a termination code in each table comprises all 1 bits.

Further, to reduce space required for decompression tables, and ensure no sequence of 32 1 bits, each code is preferably decoded as follows:

a) The first 7 bits are used to index into a table. If the character code is no more than 7 bits, it can be read directly;

b) otherwise, some number N of initial bits is discarded and the next 7 bits are used to index a second table to find the character.

Based on these steps, preferably, no character code can use more than 14 bits and all codes of more than 7 bits must fit into the code space of the N initial bits. If N is 3, for instance, then no code can use more than 10 bits.

To achieve this, the code space required for all optimal codes of more than 7 bits is first determined, following by a determining the initial offset N. Every code comprising more than N+7 bits is preferably shortened, and other codes are lengthened to balance the code tree. It is possible that this may cause the code space for codes over 7 bits to increase so that N may need to be decreased. Preferably, this process is performed in a manner that causes minimal reduction in the efficiency of the codes.

The above modifications to convention optimal algorithm yields codes in which no non-termination code ends in more than 7 1 bits, no non-termination code begins with more than 6 1 bits, no termination code is more than 14 1 bits and no

non-termination packet start code begins with more than 5 1 bits. Thus, in the middle of a packet, a sequence of no more than 13 bits of logic 1 can occur, while, at the end of a packet, a sequence of no more than 26 bits of logic 1 can occur.

In another embodiment of the present invention, Arithmetic compression can be used instead of Huffman encoding. The tables for Arithmetic encoding are preferably constructed such that a sequence of 32 bits of logic 1 will not occur in the interior of a message (which is important for a random sign-on in the middle of the stream).

Arithmetic compression provides an advantage of about 6% better compression than Huffman and uses half as much memory for tables, which allows the number of tables to be increased). Indeed, the addition of more tables and/or another level of tables yields more efficient compression. Although Arithmetic compression may take about 6 times as long as Huffman, this can certainly be improved by flattening the subroutine call tree (wherein there is a subroutine call for each output bit.)

In summary, a compression scheme according to one aspect of the invention utilizes a state machine, wherein in each state, there is a compression/decompression table comprising information on how to encode/decode the next character, as well as pointers that indicated which state to go to based on that character. A skeleton of the state machine (nodes and pointers) is preferably built by finding sequences of characters that appear often in the input. Once the skeleton has been determined, a large set of data is run through the system and counts are kept of characters seen in each state. These counts are then used to construct the encode/decode tables for the statistical compression.

Other approaches may be used to build the skeleton of the state machine. A very large fraction of the traffic on a certain feed consists of messages in the digital data feed format, which is fairly constrained. It may be possible to build by hand a skeleton that takes into account this format. For instance, capital letters only appear in the symbol name at the beginning. This long-range context information can be represented with our current approach. Once a basic skeleton is in place, the structure could be extended for sequences that occur frequently.

The above-described statistical compression schemes provide content-dependent compression and decompression. In other words, for a given data stream, the above schemes are preferably structured based on the data model associated with the given stream. It is to be appreciated, however, that other compression schemes may be employed for providing accelerated data transmission in accordance with the present invention for providing effectively increased communication bandwidth and/or reduction in latency. For instance, in another embodiment of the present invention, the data compression/decompression techniques disclosed in the above-incorporated U.S. Pat. No. 6,195,024, entitled "Content Independent Data Compression Method and System" may be used in addition to, or in lieu of, the statistical based compression schemes described above.

In general, a content-independent data compression system is a data compression system that provides an optimal compression ratio for an encoded stream regardless of the data content of the input data stream. A content-independent data compression method generally comprises the steps of compressing an input data stream, which comprises a plurality of disparate data types, using a plurality of different encoders. In other words, each encoder compresses the input data stream and outputs blocks of compressed data. An encoded data stream is then generated by selectively combining compressed data blocks output from the encoders based on compression ratios obtained by the encoders. Because a multitude of different data types may be present within a given input data stream, or data block, to it is often difficult and/or impractical to predict the level of compression that will be achieved by any one encoding technique. Indeed, rather than having to first identify the different data types (e.g., ASCII, image data, multimedia data, signed and unsigned integers, pointers, etc.) comprising an input data stream and selecting a data encoding technique that yields the highest compression ratio for each of the identified data types, content-independent data compression advantageously applies the input data stream to each of a plurality of different encoders to, in effect, generate a plurality of encoded data streams. The plurality of encoders are preferably selected based on their ability to effectively encode different types of input data. Ultimately, the final compressed data stream is generated by selectively combining blocks of the compressed streams output from the plurality of encoders. Thus, the resulting compressed output stream will achieve the greatest possible compression, regardless of the data content.

In accordance with another embodiment of the present invention, a compression system may employ both a content-dependent scheme and a content-independent scheme, such as disclosed in the above-incorporated application Ser. No. 10/016,355. In this embodiment, the content-dependent scheme is used as the primary compression/decompression system and the content-independent scheme is used in place of, or in conjunction with, the content dependent scheme, when periodically checked "compression factor" meets a predetermined threshold. For instance, the compression factor may comprise a compression ratio, wherein the compression scheme will be modified when the compression ratio falls below a certain threshold. Further, the "compression factor" may comprise the latency of data transmission, wherein the data compression scheme with be modified when the latency of data transmission exceeds a predetermined threshold.

Indeed, as explained above, the efficiency of the content-dependent compression/decompression schemes described herein is achieved, e.g., by virtue of the fact that the encoding tables are based on, and specifically designed for, the known data model. However, in situations where the data model is may be modified, the efficiency of the content-dependent scheme may be adversely affected, thereby possibly resulting in a reduction in compression efficiency and/or an increase in the overall latency of data transmission. In such a situation, as a backup system, the data compression controller can switch to a content-independent scheme that provides improved compression efficiency and reduction in latency as compared to the primary content-dependent scheme.

In yet another embodiment of the present invention, when the efficiency of a content-dependent scheme falls below a predetermined threshold based on, e.g., a change in the data structure of the data stream, the present invention preferably comprises an automatic mechanism to adaptively modify the encoding tables to generate optimal encoding tables (using the process described above with reference to FIG. **3**).

FIG. **5** is a detailed block diagram illustrates an exemplary content-independent data compression system **110** that may be employed herein. Details of this data compression system are provided in U.S. Pat. No. 6,195,024, which is fully incorporated herein by reference. In this embodiment, the data compression system **110** accepts data blocks from an input data stream and stores the input data block in an input buffer or cache **115**. It is to be understood that the system processes the input data stream in data blocks that may range in size from individual bits through complete files or collections of multiple files. Additionally, the input data block size may be

fixed or variable. A counter **120** counts or otherwise enumerates the size of input data block in any convenient units including bits, bytes, words, and double words. It should be noted that the input buffer **115** and counter **120** are not required elements of the present invention. The input data buffer **115** may be provided for buffering the input data stream in order to output an uncompressed data stream in the event that, as discussed in further detail below, every encoder fails to achieve a level of compression that exceeds an a priori specified minimum compression ratio threshold.

Data compression is performed by an encoder module **125** that may comprise a set of encoders E**1**, E**2**, E**3** . . . En. The encoder set E**1**, E**2**, E**3** . . . En may include any number "n" (where n may=1) of those lossless encoding techniques currently well known within the art such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. It is to be understood that the encoding techniques are selected based upon their ability to effectively encode different types of input data. It is to be appreciated that a full complement of encoders are preferably selected to provide a broad coverage of existing and future data types.

The encoder module **125** successively receives as input each of the buffered input data blocks (or unbuffered input data blocks from the counter module **120**). Data compression is performed by the encoder module **125** wherein each of the encoders E**1** . . . En processes a given input data block and outputs a corresponding set of encoded data blocks. It is to be appreciated that the system affords a user the option to enable/disable any one or more of the encoders E**1** . . . En prior to operation. As is understood by those skilled in the art, such feature allows the user to tailor the operation of the data compression system for specific applications. It is to be further appreciated that the encoding process may be performed either in parallel or sequentially. In particular, the encoders E**1** through En of encoder module **125** may operate in parallel (i.e., simultaneously processing a given input data block by utilizing task multiplexing on a single central processor, via dedicated hardware, by executing on a plurality of processor or dedicated hardware systems, or any combination thereof). In addition, encoders E**1** through En may operate sequentially on a given unbuffered or buffered input data block. This process is intended to eliminate the complexity and additional processing overhead associated with multiplexing concurrent encoding techniques on a single central processor and/or dedicated hardware, set of central processors and/or dedicated hardware, or any achievable combination. It is to be further appreciated that encoders of the identical type may be applied in parallel to enhance encoding speed. For instance, encoder E**1** may comprise two parallel Huffman encoders for parallel processing of an input data block.

A buffer/counter module **130** is operatively connected to the encoder module **125** for buffering and counting the size of each of the encoded data blocks output from encoder module **125**. Specifically, the buffer/counter **130** comprises a plurality of buffer/counters BC**1**, BC**2**, BC**3** . . . BCn, each operatively associated with a corresponding one of the encoders E**1** . . . En. A compression ratio module **135**, operatively connected to the output buffer/counter **130**, determines the compression ratio obtained for each of the enabled encoders E**1** . . . En by taking the ratio of the size of the input data block to the size of the output data block stored in the corresponding buffer/counters BC**1** . . . BCn. In addition, the compression ratio module **135** compares each compression ratio with an a priori-specified compression ratio threshold limit to determine if at least one of the encoded data blocks output from the enabled encoders E**1** . . . En achieves a compression that exceeds an a priori-specified threshold. As is understood by those skilled in the art, the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. A description module **138**, operatively coupled to the compression ratio module **135**, appends a corresponding compression type descriptor to each encoded data block which is selected for output so as to indicate the type of compression format of the encoded data block. A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing or transmittal. If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit, then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto. A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the input data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing or transmittal.

Again, it is to be understood that the embodiment of the data compression engine of FIG. **5** is exemplary of a preferred compression system which may be implemented in the present invention, and that other compression systems and methods known to those skilled in the art may be employed for providing accelerated data transmission in accordance with the teachings herein. Indeed, in another embodiment of the compression system disclosed in the above-incorporated U.S. Pat. No. 6,195,024, a timer is included to measure the time elapsed during the encoding process against an a priori-specified time limit. When the time limit expires, only the data output from those encoders (in the encoder module **125**) that have completed the present encoding cycle are compared to determine the encoded data with the highest compression ratio. The time limit ensures that the real-time or pseudo real-time nature of the data encoding is preserved. In addition, the results from each encoder in the encoder module **125** may be buffered to allow additional encoders to be sequentially applied to the output of the previous encoder, yielding a more optimal lossless data compression ratio. Such techniques are discussed in greater detail in the above-incorporated U.S. Pat. No. 6,195,024.

Referring now to FIG. **6**, a detailed block diagram illustrates an exemplary decompression system that may be employed herein or accelerated data transmission as disclosed in the above-incorporated U.S. Pat. No. 6,195,024. In this embodiment, the data compression engine **180** accepts compressed data blocks received over a communication channel. The decompression system processes the input data stream in data blocks that may range in size from individual bits through complete files or collections of multiple files. Additionally, the input data block size may be fixed or variable.

The data decompression engine **180** comprises an input buffer **155** that receives as input an uncompressed or compressed data stream comprising one or more data blocks. The data blocks may range in size from individual bits through complete files or collections of multiple files. Additionally,

the data block size may be fixed or variable. The input data buffer **55** is preferably included (not required) to provide storage of input data for various hardware implementations. A descriptor extraction module **160** receives the buffered (or unbuffered) input data block and then parses, lexically, syntactically, or otherwise analyzes the input data block using methods known by those skilled in the art to extract the data compression type descriptor associated with the data block. The data compression type descriptor may possess values corresponding to null (no encoding applied), a single applied encoding technique, or multiple encoding techniques applied in a specific or random order (in accordance with the data compression system embodiments and methods discussed above).

A decoder module **165** includes one or more decoders D**1** . . . Dn for decoding the input data block using a decoder, set of decoders, or a sequential set of decoders corresponding to the extracted compression type descriptor. The decoders D**1** . . . Dn may include those lossless encoding techniques currently well known within the art, including: run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. Decoding techniques are selected based upon their ability to effectively decode the various different types of encoded input data generated by the data compression systems described above or originating from any other desired source.

As with the data compression systems discussed in the above-incorporated U.S. Pat. No. 6,195,024, the decoder module **165** may include multiple decoders of the same type applied in parallel so as to reduce the data decoding time. An output data buffer or cache **170** may be included for buffering the decoded data block output from the decoder module **165**. The output buffer **70** then provides data to the output data stream. It is to be appreciated by those skilled in the art that the data compression system **180** may also include an input data counter and output data counter operatively coupled to the input and output, respectively, of the decoder module **165**. In this manner, the compressed and corresponding decompressed data block may be counted to ensure that sufficient decompression is obtained for the input data block.

Again, it is to be understood that the embodiment of the data decompression system **180** of FIG. **6** is exemplary of a preferred decompression system and method which may be implemented in the present invention, and that other data decompression systems and methods known to those skilled in the art may be employed for providing accelerated data transmission in accordance with the teachings herein.

It is to be appreciated that a data transmission acceleration system according to the present invention offers a business model by which market data vendors and users in the financial information services industry can receive various benefits. For example, the present invention affords transparent multiplication of bandwidth with minimal latency. Experiments have shown that increased bandwidth of up to 3 times can be achieved with minimal latency. Furthermore, proprietary hardware, including chip and board designs, as well as custom embedded and application software and algorithms associated with accelerated data transmission provide a cost-effective solution that can be seamlessly integrated with existing products and infrastructure. Moreover, the data acceleration through "real-time" compression and decompression affords a dramatic reduction in ongoing bandwidth costs. Further, the present invention provides mechanism to differentiate data feeds from other vendors via enriched content or quantity of the data feed.

In addition, a data compression scheme according to the present invention provides dramatically more secure and encrypted feed from current levels, thus, providing the ability to employ a secure and accelerated virtual private network over the Internet for authorized subscribers or clients with proprietary hardware and software installed.

Moreover, the present invention offers the ability to reduce a client's ongoing monthly bandwidth costs as an incentive to subscribe to a vendor's data feed service.

The present invention is readily extendable for use on a global computer network such as the Internet. This is significant since it creates a virtual private network and is important for the market data vendors and others due to its reduced cost in closed network/bandwidth solutions. In effect, the data vendors get to "ride for free" over the world's infrastructure, while still providing the same (and enhanced) services to their customers.

In yet another embodiment of the present invention a highly optimized data compression and decompression system is utilized to accelerate data transfers for data transmission feeds. This type of compression achieves very high compression ratios (over 10:1) on financial data feeds such as Nasdaq Quote Dissemination Service Data (NQDS) and SuperMontage Services. The information utilized to develop the methods described herein for Nasdaq has been garnered solely from public knowledge through specifications available from the Nasdaq Trader and Nasdaq websites. The techniques disclosed herein are broadly applicable to all financial data feeds and information or trading services.

Three types of encoding are utilized dependent upon the data fields and packet structure. In the event that a data field is unrecognizable then content independent data compression is preferably used, as previously discussed herein.

Variable Length Encoding

The basic unit of the compression process is the code. Each message field or set of set of fields being compressed together is assigned one or more codes in the range 0 . . . N. The code for a single character field is the ASCII value of the field minus 32 since all characters are in the range 32 to 127.

For various reasons, additional (escape) codes may be added to those for field values. For example, the category field has an escape code to indicate the end of a block and another to allow encoding of messages, which do not match the current format.

A basic technique used is variable rate encoding of symbols. In this approach, different amounts of the output bits are used to transmit the codes within a set. Higher frequency codes use less output bits while lower frequency codes use more output bits. Thus the average number of bits is reduced. Two methods of accomplishing this are used. The faster method uses a variant of Huffman coding while the slower method uses a form of Arithmetic coding.

In Huffman coding, each code is represent by an integral number of bits. The code sizes are computed using the standard algorithm and then (possibly) adjusted to facilitate table driven decoding (for instance, limiting codes to at most 16 bits). In the table driven decoding method used, there is a 256 element base table and two 256 element forwarding table. At each step, the next 8 bits of the input are used to index into the base table. If the code is represented in no more than 8 bits, it will be found directly. Otherwise, there will be a forwarding entry indicating which forwarding table to use and how many input bits to discard before using the next 8 bits as an index. The entry determining the result also indicates how many bits of the input to discard before processing the next field.

In arithmetic coding, the message is essentially represented as the (approximate) product of fractions with base 16384. The numerators of the fractions are proportional to the frequencies with which the codes appear in the training data.

The number of output bits used to represent a code is the base 2 logarithm of the fraction. Thus codes which appear in almost all messages may be represented with fractions of a bit.

Single Character Codes

For arithmetic coding, all single character fields are encoded as the ASCII value—32+the number of escape codes. For Huffman coding, certain single character message fields are encoded in the same way. These include:

MM Trade Desk
Quote Condition
Inside Indicator
Quote Type

Other single character fields, which have a single value that occurs most of the time, are encoded as multiple character fields (see next). In Huffman coding the smallest representation for a code is 1 bit. By combining these fields, we may encode the most common combination of values in 1 bit for the whole set. These include:

Message Category+Message Type
Session Identifier+Originator ID
PMM+Bid Price Denominator+Ask Price Denominator (Quotes)
Inside Status+Inside Type
Inside Bid Denominator+Inside Bid MC
Inside Ask Denominator+Inside Ask MC
UPC Indicator+Short Sale Bid Tick
Market of Origin+Reason

Small Set Multiple Character Codes

Multiple character fields with a small number of common values and certain combinations of single character fields are encoded based on the frequency of the combinations. A list of common combinations is used together with an escape code.

The common combinations are encoded using the corresponding code. All other combinations are encoded by the escape code followed by the (7 bit) ASCII values for the characters in the combination. The fields include the field sets above for Huffman coding as well as the following for both approaches:

Retransmission Requester
MM Location
Currency Code

Large Set Multiple Character Codes

Multiple character alphabetic or alphanumeric fields for which a large number of values are possible (Issue Symbol and MMID/MPID) are encoded as follows. Trailing spaces for Issue Symbols are deleted. Then the result is encoded using:

Variable length codes for a list of the most common values together with escapes for the possible lengths of values not in the list.

A table for the first character of the field.

A table for subsequent characters in the field.

If a value is in the list of most common values, it is encoded with the corresponding code. Otherwise, the value is encoded by sending the escape code corresponding to the (truncated) length of the value, followed by the code for the first character, which is then followed by codes for the remaining characters.

Absolute Numeric Values

Numeric fields are transmitted by sending a variable length code for the number of significant bits of the value followed by the bits of the value other than the most significant bit (which is implicitly 1). For example, 27 (a 5 bit value) would be represented by the code for a 5 bit value followed by the 4 least significant bits (11). These fields include:

Short Bid Price
Long Bid Price
Short Bid Size
Long Bid Size
Short Ask Size
Long Ask Size
Short Inside Bid Size
Long Inside Bid Size
Short Inside Ask Size
Long Inside Ask Size

Relative Numeric Values

Numeric fields expected to be close to the value of numeric values occurring earlier in the message are encoded by encoding the difference between the new value and the base value as follows:

If the difference in non-negative and less than ⅛ of the base value, the difference is encoded by sending a variable length code for the number of significant bits of the difference followed by the bits of the difference other than the most significant bit (which is implicitly 1). Otherwise, the new value is encoded by sending a variable length code for the number of significant bits of the value followed by the bits of the value other than the most significant bit (which is implicitly 1). The difference significant bit codes and the value significant bit codes are mutually exclusive. The following fields are encoded using the difference compared to the field in parentheses:

Short Ask Price (Bid Price)
Long Ask Price (Bid Price)
Short Inside Bid Price (Bid Price)
Short Inside Ask Price (Inside Bid Price)
Long Inside Bid Price (Bid Price)
Long Inside Ask Price (Inside Bid Price)

Differences

Both time and Message Sequence Number are encoded as the difference between the new value and a previous value within the compression block. This is transmitted using a code giving the sign of the difference and the number of significant bits in the absolute value of the difference followed by the bits of the absolute value other than the first.

Date

Each message within a compression block is expected to have the same date. The base date is transmitted at the beginning of the block as 7 bits of year, 4 bits of month and 5 bits of day of the month. If the date of a message is different than that of the block, a special escape code is used in place of the encoding of the sequence number and time. This is followed by the year, month and day as above followed by the time in seconds (17 bits) and the sequence number (24 bits).

Message Sequence Number and Time

Message time is converted to seconds after midnight. For all retransmitted messages (Retransmission Requester not "O"), the time is transmitted as a 17-bit value followed by the Message Sequence Number transmitted as a 24-bit value. If the date is not the same as the block date, a time value of 0x1ffff is used as an escape code.

For the first original transmission message in a block, the Message Sequence Number and time are transmitted in the same way.

For arithmetic coding of all other original transmission messages in a block, the Message Sequence Number is transmitted as the encoded change from the Message Sequence Number of the preceding original transmission message. Similarly, the time of all other original transmission messages is encoded as the difference from the previous original transmission message. An escape code in the Message Sequence Number Difference Table is used to indicate that the date is not the same as the block date.

Since almost all sequence number changes are 1 and almost all time changes are 0, we can save a bit (while Huffman coding) by encoding time and sequence number together.

This is done as follows: The most common values for both time and sequence number changes are 0 and 1 so there are three possibilities for each: 0, 1 and something else. Together this yields nine possibilities. An escape code is added to indicate a date different from the block date. To transmit the sequence number and time, the code corresponding the correct combination is first sent and then, if the time difference is not 0 or 1, the difference code for time followed by the difference code for sequence number (if required) is sent.

Unexpected Message Types

For administrative messages or non-control messages of unexpected category or type, the body of the message (the part after the header) is encoded as a 10-bit length field followed by the characters of the body encoded as 7-bit ASCII. Any Quotation message with an unexpected Inside Indicator value will have the remainder of the message encoded similarly.

Termination Code and Error Detection

Each compression block is terminated by an escape code of the message header category or category-type table. If this code is not found before the end of the block or if it is found too soon in the block, an error is returned. It is highly unlikely that a transmission error in the compressed packet could result in decoding so as to end at the same place as the original. The exception to this would be errors in transmitting bits values such as date, time or sequence number or the least significant bits of encoded values or changes. For additional error detection, a CRC check for the original could be added to compressed block.

Experimental Results

The aforecited Data Acceleration Methods were successfully applied to data captured on NASDAQ's NQDS feed. The data captured was first analyzed to optimize the Data Acceleration Methods. Essentially two distinct data rates were evaluated; one similar to the upcoming NASDAQ SuperMontage rate of 9.0 Megabits/sec and the second being the maximum data rate of the NQDS feed of 221 Kilobits/sec. In addition, two modes of data acceleration were applied—one utilizing Arithmetic and the other utilizing Huffman techniques.

The Arithmetic routines typically use 40% more CPU time than the Huffman routines and achieve approximately 15% better compression. On average the compression ratio for the SuperMontage data rate (9.0 Megabits/sec) utilizing Arithmetic Mode, yielded a value of 9.528 with a latency under 10.0 ms. This effectively says that the NQDS feed operating at a SuperMontage rate could be transmitted over one T1 line! Further overall latency can be reduced from 500 msec to something approaching 10 milliseconds if routing delays are reduced. Since the amount of data is substantially less, it will be easier and much more cost efficient to reduce routing delays. Further, since the quantity of transmitted bits is substantially smaller, the skew amongst transmitted packets will also be proportionately lower.

The average compression ratio for the standard NQDS data rate (221 Kbits/sec) was 9.3925 for the Arithmetic Mode with a latency under 128 ms. The higher latency is due to the time required to accumulated data for blocking. Since the present invention allows for very high compression ratios with small blocks of data, the latency can be reduced substantially from 128 msec without a loss in compression ratio. This effectively says that the existing NQDS feed could be transmitted over one-half of a 56 Kilobit/sec modem line. Other advantages of using data acceleration according to the invention is that such methods inherently provide (i) a high level of encryption associated with the Arithmetic Mode (with no subsequent impact on latency) and (ii) error detection capability of the decompression methods at the end user site. The first benefit produces additional levels of security for the transmitted data and the second benefit guarantees that corrupted data will not be displayed at the end user site. Furthermore, the need to dynamically compare the redundant data feeds at the end user site is eliminated.

In yet another embodiment of the present invention the aforecited algorithms and all other data compression/decompression algorithms may be utilized in a data field specific compiler that is utilized to create new data feed and data stream specific compression algorithms.

A data field description language is utilized to define a list of possible data fields and parameters along with associated data compression encoders and parameter lists. In one embodiment of the invention the data fields are defined utilizing the following convention:

```
<start list>
<list file name (optional)>
<data field a descriptor, optional parameters>
[data field a compression algorithm x, optional parameters]
<data field b descriptor, optional parameters>
[data field b compression algorithm y, optional parameters]
. . .
<data field m descriptor, optional parameters>
[data field m compression algorithm n, optional parameters]
<end list>
```

Thus start list and end list are reserved identifiers however any suitable nomenclature can be utilized.

In this simple embodiment of the present invention the list is then submitted to a data compression compiler that accepts the data field list and creates two output files. The first is a data compression algorithm set comprised of data field specific encoders and the second output file is a data decompression algorithm set comprised of encoded data field specific decoders. In practice this compiler can be implemented in any high level language, machine code, or any variant in between. In addition the language can be Java, r Visual Basic, or another interpreted language to be dynamically operated over the Internet.

More advanced embodiments of the list can be created where the order of the data fields is important to the selection of encoders. In this case the fields are an ordered vector set and the encoders are also an ordered vector set.

```
<start list>
<list file name (optional)>
<ordered data field list 1, optional parameters>
<data field a, optional parameters; data field b, optional
parameters; . . . ; data field n, optional parameters;>
[data field a compression algorithm x, optional parameters;
data field b compression algorithm y, optional
parameters; . . . ; data field m compression algorithm n]
[data field b compression algorithm x, optional parameters;
data field a compression algorithm y, optional
parameters; . . . ; data field m compression algorithm n]
<end list>
```

In this more sophisticated embodiment the encoders are selected based upon the data fields and their specific ordering.

In yet another embodiment of the present invention the sets of ordered data fields can be assigned to sets by set name, giving the ability for nesting of sets to facilitate ease of coding.

In yet another embodiment of the present invention the optional parameters to each encoder are utilized to share parameters amongst the same or different data fields.

Although illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present invention is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention. All such changes and modifications are intended to be included within the scope of the invention as defined by the appended claims.

What is claimed is:

1. A method for compressing data, wherein one or more types of encoding are applied to a data stream depending on identifiable data fields in the data stream, the method comprising:

recognizing a data field type of a data field in the data stream, wherein the data field is included in a packet;

selecting an encoder associated with the recognized data field type;

encoding the data in the data field with the selected encoder; and

providing a descriptor with the encoded data which identifies the selected encoder.

2. The method of claim **1**, wherein selecting an encoder associated with the recognized data field type is further based on a packet type of the packet.

3. The method of claim **2**, further comprising using a packet independent encoder for encoding a UDP (User Datagram Protocol) data packet.

4. The method of claim **2**, further comprising selecting packet independent or dependent encoders based on a packet type.

5. A method for compressing data, wherein one or more types of encoding are applied to a data stream depending on identifiable data fields in the data stream, the method comprising:

recognizing a data field type of a data field in a data stream;

selecting an encoder associated with the recognized data field type;

encoding the data in the data field with the selected encoder;

providing a descriptor with the encoded data which identifies the selected encoder; and

compressing data of an unrecognized data field using content independent data compression.

6. A method for compressing data, wherein one or more types of encoding are applied to a data stream depending on identifiable data fields in the data stream, the method comprising:

recognizing a data field type of a data field in a data stream;

selecting an encoder associated with the recognized data field type;

encoding the data in the data field with the selected encoder;

providing a descriptor with the encoded data which identifies the selected encoder; and

compressing data of an unrecognized packet type using content independent data compression.

7. The method of claim **1**, further comprising using arithmetic encoding with single character codes to encode data of single character fields.

8. The method of claim **1**, further comprising using Huffman encoding with small set multiple character codes to encode data of multiple character fields.

9. The method of claim **1**, further comprising using variable length encoding with large set multiple character codes to encode data of multiple character fields.

10. The method of claim **9**, wherein large set multiple character codes utilize truncation of trailing spaces.

11. The method of claim **1**, further comprising using table-based encoding with large set multiple character codes to encode data of multiple character fields.

12. The method of claim **1**, further comprising using variable length encoding with variable length codes to encode absolute number values.

13. The method of claim **1**, further comprising using numerical difference encoding to encode relative numeric values.

14. The method of claim **1**, further comprising encoding time data based on a difference between a previous time data.

15. The method of claim **1**, further comprising difference encoding based on a first absolute values within a same data packet for packet independent data compression.

16. The method of claim **1**, comprising the step of using difference encoding to encode a message sequence number.

17. The method of claim **1**, wherein the packet comprises messages, a date is encoded only one time in the packet if all messages in the packet have the same date.

18. The method of claim **17**, further comprising including date data as part of a message of the packet only if the message has a different date than the date.

19. The method of claim **1**, further comprising including error detection codes in a compressed data packet.

20. A method for creating a data feed dependent data compression routine, the method comprising:

creating a description file that describes one or more data fields and one or more encoders associated with each data field;

processing the description file with a data compression compiler; and

outputting an executable file that is used to process a stream of data by recognizing data field types in the data stream and applying encoders associated with the recognized data field types to encode the data stream.

21. The method of claim **20**, wherein the output file comprises a dynamic link library.

22. The method of claim **20**, wherein the executable file comprises instructions for performing data compression on unrecognized data fields.

23. The method of claim **22**, wherein encoder types that are used for the data compression on unrecognized data fields are specified in the description file.

24. A method for creating a data feed dependent data compression routine, the method comprising:
- creating a description file that describes one or more data fields and one or more encoders associated with each data field;
- processing the description file with a data compression compiler; and
- outputting source code that is used to process a stream of data by recognizing data field types in the data stream and applying encoders associated with the recognized data field types to encode the data stream.

25. The method of claim **1**, wherein the encoding is performed in a server.

26. The method of claim **1**, wherein the data stream comprises financial data.

27. The method of claim **1**, wherein the data stream comprises news data.

28. The method of claim **1**, further comprising transmitting the encoded data and the descriptor over a communications channel to a computer.

29. The method of claim **1**, further comprising transmitting the encoded data and the descriptor over a communications channel to a computer, wherein the encoding is performed on a server separate from the computer.

30. The method of claim **1**, further comprising transmitting the encoded data and the descriptor over a communications channel to a computer, wherein the encoding is performed on a server separate from the computer and the encoding of the data and the transmitting of the encoded data occurs faster than the data is able to be transmitted in an uuencoded form.

31. The method of claim **1**, wherein the encoding occurs in real time.

32. The method of claim **1**, wherein the encoding does not require packet-to-packet data dependency.

33. The method of claim **1**, wherein the encoded data is broadcasted over a communications channel.

34. The method of claim **1**, wherein the selected encoder is a Huffman encoder.

35. The method of claim **1**, wherein the selected encoder is a Lempel Ziv encoder.

36. The method of claim **1**, wherein the selected encoder is an Arithmetic encoder.

37. The method of claim **1**, wherein the encoded data is provided through a landline and then to and through a satellite system.

38. The method of claim **1**, wherein the encoded data is provided to a landline and then to and through a modem.

39. The method of claim **1**, wherein the packet is a UDP packet.

40. The method of claim **1**, further comprising compressing data of an unrecognized data field using one encoder.

41. The method of claim **1**, further comprising compressing data of an unrecognized field using a plurality of encoders.

42. A method for compressing data, wherein one or more types of encoding are applied to a data stream depending on identifiable data fields in the data stream, the method comprising:
- recognizing a data field type of a data field in the data stream, wherein the data field is included in a packet;
- selecting an encoder associated with the recognized data field type;
- encoding the data in the data field with the selected encoder;
- providing a descriptor with the encoded data which identifies the selected encoder; and
- compressing data of an unrecognized field using a plurality of encoders and determining the compression ratio associated with each one of the plurality of encoders.

43. A method for compressing data, wherein one or more types of encoding are applied to a data stream depending on identifiable data fields in the data stream, the method comprising:
- recognizing a data field type of a data field in the data stream, wherein the data field is included in a packet;
- selecting an encoder associated with the recognized data field type;
- encoding the data in the data field with the selected encoder;
- providing a descriptor with the encoded data which identifies the selected encoder; and
- compressing data of an unrecognized field using a plurality of encoders and determining the compression ratio associated with each one of the plurality of encoders, wherein a compressed data block is provided based on the determination.

44. A method for compressing data, wherein one or more types of encoding are applied to a data stream depending on identifiable data fields in the data stream, the method comprising:
- recognizing a data field type of a data field in the data stream, wherein the data field is included in a packet;
- selecting an encoder associated with the recognized data field type;
- encoding the data in the data field with the selected encoder;
- providing a descriptor with the encoded data which identifies the selected encoder; and
- compressing data of an unrecognized field using a plurality of encoders in parallel.

45. The method of claim **1**, further comprising compressing data of an unrecognized field using a plurality of encoders sequentially.

46. The method of claim **1**, wherein the encoded data comprises financial data and the encoded data is representative of a compression ratio greater than 10:1.

47. The method of claim **1**, wherein the encoded data is representative of a compression ratio greater than 10:1.

48. The method of claim **1**, wherein the encoded data is representative of a compression ratio greater than 10:1 and the encoded data is transmitted over a communications channel.

49. The method of claim **1**, wherein the encoded data includes an inside bid.

50. The method of claim **1**, wherein the latency associated with the encoding is under 10 milliseconds.

51. The method of claim **1**, wherein the latency associated with the encoding is under 128 milliseconds.

52. A method for creating a data feed dependent data compression routine, the method comprising:
- creating a description file that describes one or more data fields and one or more encoders associated with each data field;
- processing the description file; and
- outputting code that is used to process a stream of data by recognizing data field types in the data stream and applying encoders associated with the recognized data field types to encode the data stream.

53. A method comprising:
- recognizing a data field type of a data field in a financial data feed;
- selecting an encoder associated with the recognized data field type;

compressing the data in the data field with the selected encoder to provide compressed data;

broadcasting the compressed data to a plurality of systems, wherein the compressing achieves a compression ratio of over 10:1; and

utilizing a descriptor, which identifies the selected encoder, to decompress the compressed data on at least one of the plurality of systems.

54. The method of claim **53**, wherein the compressing is performed in a server.

55. The method of claim **53**, wherein the compressing occurs in real time.

56. The method of claim **53**, wherein the compressing does not require packet-to-packet data dependency.

57. The method of claim **53**, wherein the selected encoder is a Huffman encoder.

58. The method of claim **53**, wherein the selected encoder is a Lempel Ziv encoder.

59. The method of claim **53**, wherein the selected encoder is an Arithmetic encoder.

60. The method of claim **53**, wherein the encoded data is provided to a landline and then to and through a modem.

61. The method of claim **53**, wherein the data field is included in a packet and the packet is a UDP packet.

62. The method of claim **53**, further comprising compressing data of an unrecognized data field using one encoder.

63. The method of claim **53**, further comprising compressing data of an unrecognized field using a plurality of encoders.

64. A method comprising:

recognizing a data field type of a data field in a financial data feed;

selecting an encoder associated with the recognized data field type;

compressing the data in the data field with the selected encoder to provide compressed data;

broadcasting the compressed data to a plurality of systems, wherein the compressing achieves a compression ratio of over 10:1;

utilizing a descriptor, which identifies the selected encoder, to decompress the compressed data on at least one of the plurality of systems; and

compressing data of an unrecognized field using a plurality of encoders and determining the compression ratio associated with each one of the plurality of encoders.

65. A method comprising:

recognizing a data field type of a data field in a financial data feed;

selecting an encoder associated with the recognized data field type;

compressing the data in the data field with the selected encoder to provide compressed data;

broadcasting the compressed data to a plurality of systems, wherein the compressing achieves a compression ratio of over 10:1;

utilizing a descriptor, which identifies the selected encoder, to decompress the compressed data on at least one of the plurality of systems; and

compressing data of an unrecognized field using a plurality of encoders and determining the compression ratio associated with each one of the plurality of encoders, wherein a compressed data block is provided based on the determination.

66. A method comprising:

recognizing a data field type of a data field in a financial data feed;

selecting an encoder associated with the recognized data field type;

compressing the data in the data field with the selected encoder to provide compressed data;

broadcasting the compressed data to a plurality of systems, wherein the compressing achieves a compression ratio of over 10:1;

utilizing a descriptor, which identifies the selected encoder, to decompress the compressed data on at least one of the plurality of systems; and

compressing data of an unrecognized field using a plurality of encoders in parallel.

67. A method comprising:

recognizing a data field type of a data field in a financial data feed;

selecting an encoder associated with the recognized data field type;

compressing the data in the data field with the selected encoder to provide compressed data;

broadcasting the compressed data to a plurality of systems, wherein the compressing achieves a compression ratio of over 10:1;

utilizing a descriptor, which identifies the selected encoder, to decompress the cornpressed data on at least one of the plurality of systems; and

compressing data of an unrecognized field using a plurality of encoders sequentially.

68. The method of claim **53**, wherein the compressed data includes an inside bid.

69. The method of claim **53**, wherein the latency associated with the compressing is under 10 milliseconds.

70. The method of claim **53**, wherein the latency associated with the compressing is under 128 milliseconds.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,417,568 B2
APPLICATION NO. : 10/434305
DATED : August 26, 2008
INVENTOR(S) : James J. Fallon et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Cover page, item (63), “2001, said application No. 60/237,571 application No. 10/434,305.” should be --2001.--.

Cover page, item (60), --No. 60/237,571, filed October 3, 2000, and provisional application-- should be inserted after “application”.

Cover page, item (56), Other publications, Smith, T.B, et al., “Vo. 45” should be --Vol. 45--.

Column 1, line 63, “pate” should be --pant--.

Column 2, line 25, “financials” should be --financial--.

Column 2, line 28, --with-- should be inserted after “along”.

Column 3, line 20, --to-- should be inserted after “taken”.

Column 3, line 35, “algorithm” should be --algorithms--.

Column 3, line 42, “gradual however there” should be --gradual; however, there--.

Column 4, line 13, “forever increased” should be --for ever-increased--.

Column 4, line 27, “a 150” should be --at 150--.

Column 6, line 28, “fast” should be --faster--.

Column 6, line 40, --of-- should be inserted after “capable”.

Column 6, line 54, “decompression,” should be --decompression--.

Column 6, EQ[1], “Transmit   Accelerated” should be --Transmit Accelerated--.

Column 7, line 29, “data (is” should be --data) is--.

Column 8, line 22, “that” should be --which--.

Column 8, line 25, “and” should be --any--.

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,417,568 B2
APPLICATION NO. : 10/434305
DATED : August 26, 2008
INVENTOR(S) : James J. Fallon et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 9, line 32, “one” should be --on--.

Column 9, line 37, “decode” should be --decoding--.

Column 9, line 58, “very” should be --more--.

Column 12, line 6, “bed” should be --bcd--.

Column 12, line 35, --a-- should be inserted after “is”.

Column 12, line 56, “following by a” should be --followed by--.

Column 12, line 64, “to convention” should be --to a conventional--.

Column 13, line 13, “increased)” should be --increased--.

Column 14, line 3, “to” should be deleted.

Column 14, line 35, “with” should be --will--.

Column 14, line 42, “is” should be deleted.

Column 14, line 57, --which-- should be inserted after “diagram”.

Column 18, line 34, second occurrence of “set of” should be deleted.

Column 18, line 51, “represent” should be --represented--.

Column 20, line 21, “in” should be --is--.

Column 21, line 19, --to-- should be inserted after “corresponding”.

Column 22, line 9, “accumulated” should be --accumulate--.

Column 22, line 50, “idenitifers however” should be --identifiers; however,--.

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,417,568 B2

APPLICATION NO. : 10/434305
DATED : August 26, 2008
INVENTOR(S) : James J. Fallon et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Claim 16, column 24, line 39, “comprising the step of” should be --further comprising--.

Claim 30, column 25, line 30, “uuencoded” should be --unencoded--.

Claim 67, column 28, line 40, “cornpressed” should be --compressed--.

Signed and Sealed this

Sixth Day of January, 2009

Jon W. Dudas

JON W. DUDAS
Director of the United States Patent and Trademark Office

US007714747B2

(12) **United States Patent**
Fallon

(10) **Patent No.: US 7,714,747 B2**
(45) **Date of Patent: May 11, 2010**

(54) **DATA COMPRESSION SYSTEMS AND METHODS**

(75) Inventor: **James J. Fallon**, Armonk, NY (US)

(73) Assignee: **Realtime Data LLC**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **11/651,365**

(22) Filed: **Jan. 8, 2007**

(65) **Prior Publication Data**

US 2007/0109155 A1 May 17, 2007

Related U.S. Application Data

(63) Continuation of application No. 10/668,768, filed on Sep. 22, 2003, now Pat. No. 7,161,506, which is a continuation of application No. 10/016,355, filed on Oct. 29, 2001, now Pat. No. 6,624,761, which is a continuation-in-part of application No. 09/705,446, filed on Nov. 3, 2000, now Pat. No. 6,309,424, which is a continuation of application No. 09/210,491, filed on Dec. 11, 1998, now Pat. No. 6,195,024.

(51) **Int. Cl.**
H03M 7/34 (2006.01)

(52) **U.S. Cl.** **341/51**; 341/50; 341/67; 341/75; 341/79

(58) **Field of Classification Search** 341/50, 341/51, 67, 75, 79
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,394,352	A	7/1968	Wernikoff et al.
3,490,690	A	1/1970	Apple et al.
4,021,782	A	5/1977	Hoerning
4,032,893	A	6/1977	Moran
4,054,951	A	10/1977	Jackson et al.
4,302,775	A	11/1981	Widergren et al.
4,325,085	A	4/1982	Gooch
4,360,840	A	11/1982	Wolfrum et al.

(Continued)

FOREIGN PATENT DOCUMENTS

DE	4127518	2/1992

(Continued)

OTHER PUBLICATIONS

Rice, Robert F., "Some Practical Universal Noiseless Coding Techniques", Jet Propulsion Laboratory, Pasadena, California, JPL Publication 79-22, Mar. 15, 1979.

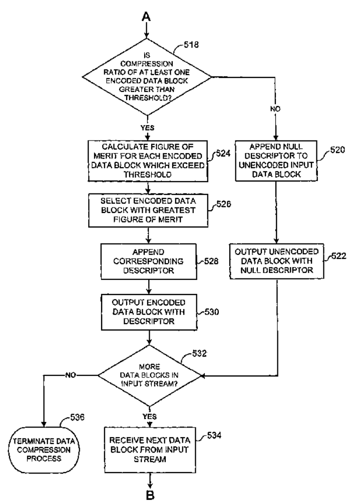
(Continued)

Primary Examiner—Linh V Nguyen
(74) *Attorney, Agent, or Firm*—Sterne, Kessler, Goldstein & Fox, PLLC

(57) **ABSTRACT**

Systems and methods for providing fast and efficient data compression using a combination of content independent data compression and content dependent data compression. In one aspect, a method for compressing data comprises the steps of: analyzing a data block of an input data stream to identify a data type of the block, the input data stream comprising a plurality of disparate data type; performing content dependent data compression on the data block, if the data type of the data block is identified; performing content independent data compression on the data block, if the data type of the data block is not identified.

22 Claims, 34 Drawing Sheets



U.S. PATENT DOCUMENTS

4,386,416	A	5/1983	Giltner et al.
4,394,774	A	7/1983	Widergren et al.
4,494,108	A	1/1985	Langdon, Jr. et al.
4,499,499	A	2/1985	Brickman et al.
4,574,351	A	3/1986	Dang et al.
4,593,324	A	6/1986	Ohkubo et al.
4,626,829	A	12/1986	Hauck
4,682,150	A	7/1987	Mathes et al.
4,701,745	A	10/1987	Waterworth
4,729,020	A	3/1988	Schaphorst et al.
4,730,348	A	3/1988	MacCrisken
4,748,638	A	5/1988	Friedman et al.
4,754,351	A	6/1988	Wright
4,804,959	A	2/1989	Makansi et al.
4,813,040	A	3/1989	Futato
4,862,167	A	8/1989	Copeland, III
4,866,601	A	9/1989	DuLac et al.
4,870,415	A	9/1989	Van Maren et al.
4,872,009	A	10/1989	Tsukiyama et al.
4,876,541	A	10/1989	Storer
4,888,812	A	12/1989	Dinan et al.
4,890,282	A	12/1989	Lambert et al.
4,897,717	A	1/1990	Hamilton et al.
4,906,991	A	3/1990	Fiala et al.
4,906,995	A	3/1990	Swanson
4,929,946	A	5/1990	O'Brien et al.
4,953,324	A	9/1990	Herrmann
4,956,808	A	9/1990	Aakre et al.
4,965,675	A	10/1990	Hori et al.
4,988,998	A	1/1991	O'Brien
5,003,307	A	3/1991	Whiting et al.
5,016,009	A	5/1991	Whiting et al.
5,028,922	A	7/1991	Huang
5,045,848	A	9/1991	Fascenda
5,045,852	A	9/1991	Mitchell et al.
5,046,027	A	9/1991	Taaffe et al.
5,049,881	A	9/1991	Gibson et al.
5,091,782	A	2/1992	Krause et al.
5,097,261	A	3/1992	Langdon, Jr. et al.
5,109,226	A	4/1992	Mac Lean, Jr. et al.
5,109,433	A	4/1992	Notenboom
5,113,522	A	5/1992	Dinwiddie, Jr. et al.
5,115,309	A	5/1992	Hang
5,121,342	A	6/1992	Szymborski et al.
5,126,739	A	6/1992	Whiting et al.
5,128,963	A	7/1992	Akagiri
5,146,221	A	9/1992	Whiting et al.
5,150,430	A	9/1992	Chu
5,155,484	A	10/1992	Chambers, IV
5,159,336	A	10/1992	Rabin et al.
5,167,034	A	11/1992	MacLean, Jr. et al.
5,175,543	A	12/1992	Lantz
5,179,651	A	1/1993	Taaffe et al.
5,187,793	A	2/1993	Keith et al.
5,191,431	A	3/1993	Hasegawa et al.
5,204,756	A	4/1993	Chevion et al.
5,209,220	A	5/1993	Hiyama et al.
5,212,742	A	5/1993	Normile et al.
5,226,176	A	7/1993	Westaway et al.
5,227,893	A	7/1993	Ett
5,231,492	A	7/1993	Dangi et al.
5,237,460	A	8/1993	Miller et al.
5,237,675	A	8/1993	Hannon, Jr.
5,243,341	A	9/1993	Seroussi et al.
5,243,348	A	9/1993	Jackson
5,247,638	A	9/1993	O'Brien et al.
5,247,646	A	9/1993	Osterlund et al.
5,249,053	A	9/1993	Jain
5,263,168	A	11/1993	Toms et al.
5,270,832	A	12/1993	Balkanski et al.
5,280,600	A	1/1994	Van Maren et al.
5,287,420	A	2/1994	Barrett
5,289,580	A	2/1994	Latif et al.
5,293,379	A	3/1994	Carr
5,293,576	A	3/1994	Mihm, Jr. et al.
5,307,497	A	4/1994	Feigenbaum et al.
5,309,555	A	5/1994	Akins et al.
5,341,440	A	8/1994	Earl et al.
5,347,600	A	9/1994	Barnsley et al.
5,355,498	A	10/1994	Provino et al.
5,357,614	A	10/1994	Pattisam et al.
5,367,629	A	11/1994	Chu et al.
5,372,290	A	12/1994	Lempel et al.
5,379,036	A	1/1995	Storer
5,379,757	A	1/1995	Hiyama et al.
5,381,145	A	1/1995	Allen et al.
5,389,922	A	2/1995	Seroussi et al.
5,394,534	A	2/1995	Kulakowski et al.
5,396,228	A	3/1995	Garahi
5,400,401	A	3/1995	Wasilewski et al.
5,403,639	A	4/1995	Belsan et al.
5,406,278	A	4/1995	Graybill et al.
5,406,279	A	4/1995	Anderson et al.
5,410,671	A	4/1995	Elgamal et al.
5,412,384	A	5/1995	Chang et al.
5,414,850	A	5/1995	Whiting
5,420,639	A	5/1995	Perkins
5,434,983	A	7/1995	Yaso et al.
5,437,020	A	7/1995	Wells et al.
5,452,287	A	9/1995	DiCecco et al.
5,454,079	A	9/1995	Roper et al.
5,454,107	A	9/1995	Lehman et al.
5,455,576	A *	10/1995	Clark et al. 341/50
5,455,680	A	10/1995	Shin
5,461,679	A	10/1995	Normile et al.
5,463,390	A	10/1995	Whiting et al.
5,467,087	A	11/1995	Chu
5,471,206	A	11/1995	Allen et al.
5,479,587	A	12/1995	Campbell et al.
5,479,633	A	12/1995	Wells et al.
5,483,470	A	1/1996	Alur et al.
5,486,826	A	1/1996	Remillard
5,488,364	A	1/1996	Cole
5,495,244	A	2/1996	Jeong et al.
5,504,842	A	4/1996	Gentile
5,506,844	A	4/1996	Rao
5,506,872	A	4/1996	Mohler
5,506,944	A	4/1996	Gentile
5,528,628	A	6/1996	Park et al.
5,530,845	A	6/1996	Hiatt et al.
5,533,051	A	7/1996	James
5,535,311	A	7/1996	Zimmerman
5,535,356	A	7/1996	Kim et al.
5,535,369	A	7/1996	Wells et al.
5,537,658	A	7/1996	Bakke et al.
5,539,865	A	7/1996	Gentile
5,542,031	A	7/1996	Douglass et al.
5,544,290	A	8/1996	Gentile
5,546,395	A	8/1996	Sharma et al.
5,546,475	A	8/1996	Bolle et al.
5,553,160	A	9/1996	Dawson
5,557,551	A	9/1996	Craft
5,557,668	A	9/1996	Brady
5,557,749	A	9/1996	Norris
5,561,421	A	10/1996	Smith et al.
5,561,824	A	10/1996	Carreiro et al.
5,563,961	A	10/1996	Rynderman et al.
5,574,952	A	11/1996	Brady et al.
5,574,953	A	11/1996	Rust et al.
5,576,953	A	11/1996	Hugentobler
5,581,715	A	12/1996	Verinsky et al.
5,583,500	A	12/1996	Allen et al.
5,586,264	A	12/1996	Belknap et al.
5,586,285	A	12/1996	Hasbun et al.

5,590,306	A	12/1996	Watanabe et al.	
5,596,674	A	1/1997	Bhandari et al.	
5,604,824	A	2/1997	Chui et al.	
5,606,706	A	2/1997	Takamoto et al.	
5,610,657	A	3/1997	Zhang	
5,611,024	A	3/1997	Campbell et al.	
5,612,788	A	3/1997	Stone	
5,613,069	A	3/1997	Walker	
5,615,017	A	3/1997	Choi	
5,615,287	A	3/1997	Fu et al.	
5,619,995	A	4/1997	Lobodzinski	
5,621,820	A	4/1997	Rynderman et al.	
5,623,623	A	4/1997	Kim et al.	
5,623,701	A	4/1997	Bakke et al.	
5,627,534	A	5/1997	Craft	
5,627,995	A	5/1997	Miller et al.	
5,629,732	A	5/1997	Moskowitz et al.	
5,630,092	A	5/1997	Carreiro et al.	
5,635,632	A	6/1997	Fay et al.	
5,635,932	A	6/1997	Shinagawa et al.	
5,638,498	A	6/1997	Tyler et al.	
5,640,158	A	6/1997	Okayama et al.	
5,642,506	A	6/1997	Lee	
5,649,032	A	7/1997	Burt et al.	
5,652,795	A	7/1997	Dillon et al.	
5,652,857	A	7/1997	Shimoi et al.	
5,652,917	A	7/1997	Maupin et al.	
5,654,703	A	8/1997	Clark, II	
5,655,138	A	8/1997	Kikinis	
5,666,560	A	9/1997	Moertl et al.	
5,668,737	A	9/1997	Iler	
5,671,355	A	9/1997	Collins	
5,671,389	A	9/1997	Saliba	
5,671,413	A	9/1997	Shipman et al.	
5,675,333	A	10/1997	Boursier et al.	
5,675,789	A	10/1997	Ishii et al.	
5,686,916	A	11/1997	Bakhmutsky	
5,692,159	A	11/1997	Shand	
5,694,619	A	12/1997	Konno	
5,696,927	A	12/1997	MacDonald et al.	
5,703,793	A	12/1997	Wise et al.	
5,708,511	A	1/1998	Gandhi et al.	
5,715,477	A	2/1998	Kikinis	
5,717,393	A	2/1998	Nakano et al.	
5,717,394	A	2/1998	Schwartz et al.	
5,719,862	A	2/1998	Lee et al.	
5,721,958	A	2/1998	Kikinis	
5,724,475	A	3/1998	Kirsten	
5,729,228	A	3/1998	Franaszek et al.	
5,740,395	A	4/1998	Wells et al.	
5,742,773	A	4/1998	Blomfield-Brown et al.	
5,748,904	A	5/1998	Huang et al.	
5,757,852	A	5/1998	Jericevic et al.	
5,765,027	A	6/1998	Wang et al.	
5,767,898	A *	6/1998	Urano et al.	348/43
5,768,445	A	6/1998	Troeller et al.	
5,768,525	A	6/1998	Kralowetz et al.	
5,771,340	A	6/1998	Nakazato et al.	
5,774,715	A	6/1998	Madany et al.	
5,778,411	A	7/1998	DeMoss et al.	
5,781,767	A	7/1998	Inoue et al.	
5,784,572	A	7/1998	Rostoker et al.	
5,787,487	A	7/1998	Hashimoto et al.	
5,794,229	A	8/1998	French et al.	
5,796,864	A	8/1998	Callahan	
5,799,110	A	8/1998	Israelsen et al.	
5,805,834	A	9/1998	McKinley et al.	
5,805,932	A	9/1998	Kawashima et al.	
5,808,660	A	9/1998	Sekine et al.	
5,809,176	A	9/1998	Yajima	
5,809,299	A	9/1998	Cloutier et al.	
5,809,337	A	9/1998	Hannah et al.	
5,812,195	A	9/1998	Zhang	
5,812,789	A	9/1998	Diaz et al.	
5,818,368	A	10/1998	Langley	
5,818,369	A	10/1998	Withers	
5,818,530	A	10/1998	Canfield et al.	
5,819,215	A	10/1998	Dobson et al.	
5,822,781	A	10/1998	Wells et al.	
5,825,424	A	10/1998	Canfield et al.	
5,825,830	A	10/1998	Kopf	
5,832,037	A	11/1998	Park	
5,832,126	A	11/1998	Tanaka	
5,835,788	A	11/1998	Blumer et al.	
5,836,003	A	11/1998	Sadeh	
5,838,996	A	11/1998	deCarmo	
5,839,100	A	11/1998	Wegener	
5,841,979	A	11/1998	Schulhof et al.	
5,847,762	A	12/1998	Canfield et al.	
5,850,565	A	12/1998	Wightman	
5,861,824	A	1/1999	Ryu et al.	
5,861,920	A	1/1999	Mead et al.	
5,864,342	A	1/1999	Kajiya et al.	
5,867,167	A	2/1999	Deering	
5,867,602	A	2/1999	Zandi et al.	
5,870,036	A	2/1999	Franaszek et al.	
5,870,087	A	2/1999	Chau	
5,872,530	A	2/1999	Domyo et al.	
5,874,907	A	2/1999	Craft	
5,883,975	A	3/1999	Narita et al.	
5,884,269	A	3/1999	Cellier et al.	
5,886,655	A	3/1999	Rust	
5,887,165	A	3/1999	Martel et al.	
5,889,961	A	3/1999	Dobbek	
5,892,847	A	4/1999	Johnson	
5,909,557	A	6/1999	Betker et al.	
5,909,559	A	6/1999	So	
5,915,079	A	6/1999	Vondran, Jr. et al.	
5,917,438	A	6/1999	Ando	
5,918,068	A	6/1999	Shafe'	
5,918,225	A	6/1999	White et al.	
5,920,326	A	7/1999	Rentschler et al.	
5,923,860	A	7/1999	Olarig	
5,930,358	A	7/1999	Rao	
5,936,616	A	8/1999	Torborg, Jr. et al.	
5,943,692	A	8/1999	Marberg et al.	
5,945,933	A	8/1999	Kalkstein	
5,949,355	A	9/1999	Panaoussis	
5,951,623	A	9/1999	Reynar et al.	
5,955,976	A *	9/1999	Heath	341/87
5,956,490	A	9/1999	Buchholz et al.	
5,960,465	A	9/1999	Adams	
5,964,842	A	10/1999	Packard	
5,968,149	A	10/1999	Jaquette et al.	
5,969,927	A	10/1999	Schirmer et al.	
5,973,630	A	10/1999	Heath	
5,974,235	A	10/1999	Nunally et al.	
5,974,387	A	10/1999	Kageyama	
5,974,471	A	10/1999	Belt	
5,978,483	A	11/1999	Thompson, Jr. et al.	
5,982,360	A	11/1999	Wu et al.	
5,982,723	A	11/1999	Kamatani	
5,987,022	A	11/1999	Geiger et al.	
5,987,590	A	11/1999	So	
5,990,884	A	11/1999	Douma et al.	
5,991,515	A	11/1999	Fall et al.	
5,996,033	A	11/1999	Chiu-Hao	
6,000,009	A	12/1999	Brady	
6,002,411	A	12/1999	Dye	
6,003,115	A	12/1999	Spear et al.	
6,008,743	A	12/1999	Jaquette	
6,011,901	A	1/2000	Kirsten	
6,014,694	A	1/2000	Aharoni et al.	
6,023,755	A	2/2000	Casselman	
6,026,217	A	2/2000	Adiletta	
6,028,725	A	2/2000	Blumenau	

6,031,939	A	2/2000	Gilbert et al.	
6,032,148	A	2/2000	Wilkes	
6,058,459	A	5/2000	Owen et al.	
6,061,398	A	5/2000	Satoh et al.	
6,070,179	A	5/2000	Craft	
6,073,232	A	6/2000	Kroeker et al.	
6,075,470	A	6/2000	Little et al.	
6,078,958	A	6/2000	Echeita et al.	
6,091,777	A	7/2000	Guetz et al.	
6,092,123	A	7/2000	Steffan et al.	
6,094,634	A	7/2000	Yahagi et al.	
6,097,520	A	8/2000	Kadnier	
6,098,114	A	8/2000	McDonald et al.	
6,104,389	A	8/2000	Ando	
6,105,130	A	8/2000	Wu et al.	
6,128,412	A	10/2000	Satoh	
6,134,631	A	10/2000	Jennings, III	
6,141,053	A	10/2000	Saukkonen	
6,145,020	A	11/2000	Barnett	
6,145,069	A	11/2000	Dye	
6,169,241	B1	1/2001	Shimizu	
6,170,007	B1	1/2001	Venkatraman et al.	
6,170,047	B1	1/2001	Dye	
6,170,049	B1	1/2001	So	
6,172,936	B1	1/2001	Kitazaki	
6,173,381	B1	1/2001	Dye	
6,175,650	B1	1/2001	Sindhu et al.	
6,175,856	B1	1/2001	Riddle	
6,182,125	B1	1/2001	Borella et al.	
6,185,625	B1	2/2001	Tso et al.	
6,185,659	B1	2/2001	Milillo et al.	
6,192,082	B1	2/2001	Moriarty et al.	
6,192,155	B1	2/2001	Fan	
6,195,024	B1	2/2001	Fallon	
6,195,465	B1	2/2001	Zandi et al.	
6,208,273	B1	3/2001	Dye et al.	
6,215,904	B1	4/2001	Lavallee	
6,219,754	B1	4/2001	Belt et al.	
6,222,886	B1	4/2001	Yogeshwar	
6,225,922	B1	5/2001	Norton	
6,226,667	B1	5/2001	Matthews et al.	
6,226,740	B1	5/2001	Iga	
6,230,223	B1	5/2001	Olarig	
6,237,054	B1	5/2001	Freitag, Jr.	
6,243,829	B1	6/2001	Chan	
6,253,264	B1	6/2001	Sebastian	
6,272,178	B1 *	8/2001	Nieweglowski et al.	375/240.03
6,272,627	B1	8/2001	Mann	
6,272,628	B1	8/2001	Aguilar et al.	
6,282,641	B1	8/2001	Christensen	
6,298,408	B1	10/2001	Park	
6,308,311	B1	10/2001	Carmichael et al.	
6,309,424	B1 *	10/2001	Fallon	341/51
6,310,563	B1	10/2001	Har et al.	
6,317,714	B1	11/2001	Del Castillo et al.	
6,330,622	B1	12/2001	Schaefer	
6,345,307	B1	2/2002	Booth	
6,356,589	B1	3/2002	Gebler et al.	
6,356,937	B1	3/2002	Montville et al.	
6,392,567	B2	5/2002	Satoh	
6,404,931	B1	6/2002	Chen et al.	
6,421,387	B1	7/2002	Rhee	
6,434,168	B1	8/2002	Kari	
6,434,695	B1	8/2002	Esfahani et al.	
6,442,659	B1	8/2002	Blumenau	
6,449,658	B1	9/2002	Lafe et al.	
6,449,682	B1	9/2002	Toorians	
6,452,602	B1	9/2002	Morein	
6,452,933	B1	9/2002	Duffield et al.	
6,459,429	B1	10/2002	Deering	
6,463,509	B1	10/2002	Teoman et al.	
6,487,640	B1	11/2002	Lipasti	
6,489,902	B2	12/2002	Heath	
6,513,113	B1	1/2003	Kobayashi	
6,523,102	B1	2/2003	Dye et al.	
6,526,174	B1	2/2003	Graffagnino	
6,529,633	B1	3/2003	Easwar et al.	
6,532,121	B1	3/2003	Rust et al.	
6,539,438	B1	3/2003	Ledzius et al.	
6,539,456	B2	3/2003	Stewart	
6,542,644	B1	4/2003	Satoh	
6,577,254	B2	6/2003	Rasmussen	
6,590,609	B1	7/2003	Kitade et al.	
6,597,812	B1	7/2003	Fallon et al.	
6,601,104	B1	7/2003	Fallon	
6,604,040	B2	8/2003	Kawasaki et al.	
6,604,158	B1	8/2003	Fallon	
6,606,040	B2 *	8/2003	Abdat	341/87
6,606,413	B1	8/2003	Zeineh	
6,609,223	B1	8/2003	Wolfgang	
6,618,728	B1	9/2003	Rail	
6,624,761	B2	9/2003	Fallon	
6,650,261	B2 *	11/2003	Nelson et al.	341/106
6,661,839	B1	12/2003	Ishida et al.	
6,661,845	B1	12/2003	Herath	
6,704,840	B2	3/2004	Nalawadi et al.	
6,711,709	B1 *	3/2004	York	714/748
6,717,534	B2	4/2004	Yokose	
6,731,814	B2 *	5/2004	Zeck et al.	382/239
6,745,282	B2	6/2004	Okada et al.	
6,748,457	B2	6/2004	Fallon et al.	
6,756,922	B2	6/2004	Ossia	
6,768,749	B1	7/2004	Osler et al.	
6,792,151	B1	9/2004	Barnes et al.	
6,810,434	B2	10/2004	Muthujumaraswathy et al.	
6,813,689	B2	11/2004	Baxter, III	
6,819,271	B2	11/2004	Geiger et al.	
6,822,589	B1	11/2004	Dye et al.	
6,856,651	B2	2/2005	Singh	
6,862,278	B1	3/2005	Chang et al.	
6,879,266	B1	4/2005	Dye et al.	
6,885,316	B2	4/2005	Mehring	
6,885,319	B2	4/2005	Geiger et al.	
6,888,893	B2	5/2005	Li et al.	
6,909,383	B2	6/2005	Shokrollahi et al.	
6,944,740	B2	9/2005	Abali et al.	
6,959,359	B1	10/2005	Suzuki et al.	
6,963,608	B1	11/2005	Wu	
6,993,597	B2	1/2006	Nakagawa et al.	
7,007,099	B1	2/2006	Donati et al.	
7,054,493	B2	5/2006	Schwartz	
7,069,342	B1	6/2006	Biederman	
7,089,391	B2	8/2006	Geiger et al.	
7,102,544	B1	9/2006	Liu	
7,129,860	B2	10/2006	Alvarez, II et al.	
7,130,913	B2	10/2006	Fallon	
7,161,506	B2	1/2007	Fallon	
7,181,608	B2	2/2007	Fallon et al.	
7,190,284	B1	3/2007	Dye et al.	
7,319,667	B1	1/2008	Biederman	
7,321,937	B2	1/2008	Fallon	
7,330,912	B1	2/2008	Fox et al.	
7,352,300	B2	4/2008	Fallon	
7,358,867	B2	4/2008	Fallon	
7,376,772	B2	5/2008	Fallon	
7,378,992	B2	5/2008	Fallon	
7,386,046	B2	6/2008	Fallon	
7,395,345	B2	7/2008	Fallon	
7,400,274	B2	7/2008	Fallon	
7,415,530	B2	8/2008	Fallon	
7,417,568	B2	8/2008	Fallon et al.	
7,552,069	B2	6/2009	Kepecs	
7,565,441	B2	7/2009	Romanik et al.	
2001/0031092	A1 *	10/2001	Zeck et al.	382/239
2001/0032128	A1	10/2001	Kepecs	
2001/0047473	A1	11/2001	Fallon	

2001/0052038	A1	12/2001	Fallon et al.	
2001/0054131	A1	12/2001	Alvarez, II et al.	
2002/0037035	A1	3/2002	Singh	
2002/0080871	A1	6/2002	Fallon et al.	
2002/0101367	A1	8/2002	Geiger et al.	
2002/0104891	A1	8/2002	Otto	
2002/0126755	A1	9/2002	Li et al.	
2002/0191692	A1	12/2002	Fallon et al.	
2003/0030575	A1 *	2/2003	Frachtenberg et al.	341/51
2003/0034905	A1	2/2003	Anton et al.	
2003/0084238	A1	5/2003	Okada et al.	
2003/0142874	A1	7/2003	Schwartz	
2003/0191876	A1	10/2003	Fallon	
2004/0042506	A1	3/2004	Fallon et al.	
2004/0073710	A1	4/2004	Fallon	
2006/0015650	A1	1/2006	Fallon	
2006/0181441	A1	8/2006	Fallon	
2006/0181442	A1	8/2006	Fallon	
2006/0184696	A1	8/2006	Fallon	
2006/0190644	A1	8/2006	Fallon	
2006/0195601	A1	8/2006	Fallon	
2007/0043939	A1	2/2007	Fallon et al.	
2007/0050514	A1	3/2007	Fallon	
2007/0050515	A1	3/2007	Fallon	
2007/0067483	A1	3/2007	Fallon	
2007/0083746	A1	4/2007	Fallon et al.	
2007/0109154	A1	5/2007	Fallon	
2007/0109156	A1	5/2007	Fallon	
2007/0174209	A1	7/2007	Fallon et al.	
2008/0232457	A1	9/2008	Fallon	
2009/0154545	A1	6/2009	Fallon et al.	
2009/0287839	A1	11/2009	Fallon et al.	

FOREIGN PATENT DOCUMENTS

EP	0164677	12/1985
EP	0185098	6/1986
EP	0283798	9/1988
EP	0595406	5/1994
EP	0405572	11/1994
EP	0718751	6/1996
EP	0493130	6/1997
EP	0587437	2/2002
GB	2162025	1/1986
JP	6051989	2/1994
JP	9188009	7/1997
JP	11149376	6/1999
WO	WO 9414273	6/1994
WO	WO 9429852	12/1994
WO	WO 9502873	1/1995
WO	WO 9748212	12/1997
WO	WO 9908186	2/1999
WO	WO 02/39591	5/2002

OTHER PUBLICATIONS

Anderson, J., et al. “Codec squeezes color teleconferencing through digital telephone lines”, Electronics 1984, pp. 113-115.
Venbrux, Jack, “A VLSI Chip Set for High-Speed Lossless Data Compression”, IEEE Trans. On Circuits and Systems for Video Technology, vol. 2, No. 44, Dec. 1992, pp. 381-391.
“Fast Dos Soft Boot”, IBM Technical Disclosure Bulletin, Feb. 1994, vol. 37, Issue No. 2B, pp. 185-186.
“Operating System Platform Abstraction Method”, IBM Technical Disclosure Bulletin, Feb. 1995, vol. 38, Issue No. 2, pp. 343-344.
Murashita, K., et al., “High-Speed Statistical Compression using Self-Organized Rules and Predetermined Code Tables”, IEEE, 1996 Data Compression Conference.
Coene, W., et al. “A Fast Route For Application of Rate-distortion Optimal Quantization in an MPEG Video Encoder” Proceedings of the International Conference on Image Processing, US., New York, IEEE, Sep. 16, 1996, pp. 825-828.
Rice, Robert, “Lossless Coding Standards for Space Data Systems”, IEEE 1058-6393/97, pp. 577-585.
Millman, Howard, “Image and video compression”, Computerworld, vol. 33, Issue No. 3, Jan. 18, 1999, pp. 78.
“IBM boosts your memory”, Geek.com [online], Jun. 26, 2000 [retrieved on Jul. 6, 2007], <URL:http://www.geek.com/ibm-boosts-your-memory/>.
“IBM Research Breakthrough Doubles Computer Memory Capacity”, IBM Press Release [online], Jun. 26, 2000 [retrieved on Jul. 6, 2007], <URL:http://www-03.ibm.com/press/us/en/pressrelease/1653.wss>.
“ServerWorks To Deliver IBM’s Memory eXpansion Technology in Next-Generation Core Logic for Servers”, ServerWorks Press Release [online], Jun. 27, 2000 [retrieved on Jul. 14, 2000], <URL:http://www.serverworks.com/news/press/000627.html>.
Abali, B., et al., “Memory Expansion Technology (MXT) Software support and performance”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 287-301.
Franaszek, P. A., et al., “Algorithms and data structures for compressed-memory machines”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 245-258.
Franaszek, P. A., et al., “On internal organization in compressed random-access memories”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 259-270.
Smith, T.B., et al., “Memory Expansion Technology (MXT) Competitive impact”, IBM Journal of Research and Development, Vo. 45, Issue No. 2, Mar. 2001, pp. 303-309.
Tremaine, R. B., et al., “IBM Memory Expansion Technology (MXT)”, IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 271-285.
Yeh, Pen-Shu, “The CCSDS Lossless Data Compression Recommendation for Space Applications”, Chapter 16, Lossless Compression Handbook, Elsevier Science (USA), 2003, pp. 311-326.
Expand Networks Accelerator 4000 Series User’s Guide.
Tridgell, Andrew; “Efficient Algorithms for Sorting and Synchronization”; A thesis submitted for the degree of Doctor of Philosophy at the Australian National University; Feb. 1999; pp. iii-106.
Jung, et al.; “Performance optimization of wireless local area networks through VLSI data compression”; Wireless Networks, vol. 4, 1998; pp. 27-39.
Jones, et al.; “Lossless data compression for short duration 3D frames in positron emission tomography”, IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference; vol. 3; pp. 1831-1834.
Maier, Mark W.; “Algorithm Evaluation for the Synchronous Data Compression Standard”; University of Alabama; pp. 1-10. 1988.
Bassiouni, et al.; “A Scheme for Data Compression in Supercomputers”; IEEE; 1988; pp. 272-278.
Welch, Terry A.; “A Technique for High-Performance Data Compression”; IEEE; Jun. 1984; pp. 8-19.
ALDC: Adaptive Lossless Data Compression; IBM; 1994.
ALDC-Macro: Adaptive Lossless Data Compression; IBM Corporation; 1994.
ALDC1-20S: Adaptive Lossless Data Compression; IBM Corporation; 1994.
ALDC1-40S: Adaptive Lossless Data Compression; IBM Corporation; 1994.
ALDC1-5S: Adaptive Lossless Data Compression; IBM Corporation; 1994.
Craft, David J.; “Data Compression Choice No Easy Call”; Computer Technology Review; vol. XIV, No. 1; Jan. 1994.
Costlow, Terry; “Sony designs faster, denser tape drive”; Electronic Engineering Times; May 20, 1996, pp. 86-87.
Wilson, Ron; “IBM ups compression ante”; Electronic Engineering Times; Aug. 16, 1993; pp. 1-94.
“IBM Announces New Feature for 3480 Subsystem”; Tucson Today; vol. 12, No. 337, Jul. 25, 1989.
Syngress Media, Inc.; “CCA Citrix Certified Administrator for MetaFrame 1.8 Study Guide”; 2000.
International Telecommunication Union; “Data Compression Procedures for Data Circuit Terminating Equipment (DCE) Using Error Correction Procedures”; Geneva, 1990.
Cheng, et al.; “A fast, highly reliable data compression chip and algorithm for storage systems”; IBM J. Res. Develop.; vol. 40, No. 6, Nov. 1996; pp. 603-613.

Cisco Systems; "Cisco IOS Data Compression"; 1997; pp. 1-10.
Craft, D. J.; "A fast hardware data compression algorithm and some algorithmic extensions"; IBM J. Res. Develop.; vol. 42; No. 6; Nov. 6, 1998; pp. 733-746.
Rustici, Robert; "Enhanced CU-SeeMe"1995, Zero in Technologies, Inc.
White Pine Software; "CU-SeeMe Pro: Quick Start Guide"; Version 4.0 for Windows; 1999.
"CU-SeeMe Reflector"; www.geektimes.com/michael/CU-SeeMe/faqs/reflectors.html; accessed on Dec. 2, 2008.
Daniels, et al.; "Citrix WinFrame 1.6 Beta"; May 1, 1996; license.icopyright.net/user/downloadLicense.act?lic=3.7009-9123; accessed Dec. 2, 2008.
Held, et al.; "Data Compression"; Third Edition; John Wiley & Sons Ltd.; 1991.
Data Compression Applications and Innovations Workshop; Proceedings of a Workshop held in Conjunction with the IEEE Data Compression Conference; Snowbird, Utah; Mar. 31, 1995.
Britton, et al.; "Discovery Desktop Conferencing with NetMeeting 2.0"; IDG Books Worldwide, inc.; 1997.
Sattler, Michael; "Internet Tv with Cu-SeeMe"; Sams.Net Publishing; 1995; First Edition.
IBM Microelectronics Comdex Fall '93 Booth Location.
Disz, et al.; "Performance Model of the Argonne Voyager Multimedia Server; IEEE; 1997; pp. 316-327.
"Downloading and Installing NetMeeting"; www.w4mq.com/help/h3.htm; accessed on Dec. 2, 2008.
Fox, et al.; "Adapting to Network and Client Variability via On-Demand Dynamic Distillation"; ASPLOS VII; Oct. 1996; pp. 160-170.
Fox, et al.; "Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perceptives"; IEEE Personal Communications, Aug. 1998; pp. 10-19.
Han, et al.; "CU-SeeMe VR Immersive Desktop Teleconferencing"; Department of Computer Science; Cornell University; To appear in ACM Multimedia 1996.
Howard, et al.; "Parallel Lossless Image Compression Using Huffman and Arithmetic Coding"; 1992; pp. 1-9.
Howard, Paul G.; "Text Image Compression Using Soft Pattern Matching"; The Computer Journal; vol. 40, No. 2/3; 1997; pp. 146-156.
Howard, et al.; "The Emerging JBIG2 Standard"; IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, No. 7, Nov. 1998; pp. 838-848.
Craft, D. J.; "A fast hardware data compression algorithm and some algorithmic extensions"; Journal of Research and Development; vol. 42, No. 6, Nov. 1998; pp. 733-745.
"Direct Access Storage Device Compression and Decompression Data Flow"; IBM Technical Disclosure Bulletin; vol. 38, No. 11; Nov. 1995; pp. 291-295.
ICA Timeline, Sep. 24, 2007.
Converse, et al.; "Low Bandwidth X Extension"; Protocol Version 1.0; X Consortium; Dec. 21, 1996.
Magstar and IBM 3590 High Performance Tape Subsystem Technical Guide; Nov. 1996; IBM International Technical Support Organization.
MetaFrame Administration Student Workbook; Jun. 1998; Citrix Professional Courseware; Citrix Systems, Inc.
NCD WinCenter 3.1: Bringing Windows to Every Desktop; 1998.
Overview NetMeeting 2.1; Microsoft Tech Net; technet.microsoft.com/en-us/library/cc767141(printer).aspx; accessed Dec. 2, 2008.
NetMeeting 2.1 Resource Kit; Microsoft TechNet; technet.microsoft.com/en-us/library/cc767142(printer).aspx; accessed on Dec. 2, 2008.
Conferencing Standards: NetMeeeting 2.1 Resource Kit; Microsoft TechNet; technet.microsoft.com/en-us/library/cc767150(printer).aspx; accessed Dec. 2, 2008.
Summers, Bob; "Official Microsoft NetMeeting Book"; Microsoft Press; 1998.
Zebrose, Katherin L. "Iniegrating Hardware Accelerators into Internetworking Switches"; Telco Systems 1994.
Simpson, et al.; "A Multiple Processor Approach to Data Compression"; ACM; 1998; pp. 641-649.
"IBM Technology Products Introduces New Family of High-Performance Data Compression Products"; IBM; Aug. 16, 1993.
ReadMe; PowerQuest Drive Image Pro; Version 3.00; 1994-1999; PowerQuest Corporation; p. 1-6.
Schulzrinne, et al.; "RTP Profile for Audio and Video Conferences with Minimal Control"; Jan. 1996; www.ietf.org/rfc/rfc1890.txt; accessed on Dec. 3, 2008.
Zhu, C.; "RTP Payload Format for H.263 Video Streams"; Standards Track; Sep. 1997; pp. 1-12.
Simpson, W.; "The Point-To-Point Protocol (PPP)"; Standards Track; Jul. 1994; pp. i-52.
Reynolds, et al.; "Assigned Numbers"; Standards Track; Oct. 1994; pp. 1-230.
Deutsch, et al.; "ZLIP Compressed Data Format Specification version 3.3"; Informational; May 1996; p. 1-10.
Deutsch, P.; "DEFLATE Compressed Data Format Specification version 1.3"; Informational; May 1996; pp. 1-15.
Rand, D.; "The PPP Compression Control Protocol (CCP)"; Standards Track; Jun. 1996; pp. 1-9.
Schneider, et al.; "PPP LZS-DCP Compression Protocol (LZS-DCP)"; Informational; Aug. 1996; pp. 1-18.
Friend, et al.; "PPP Stac LZS Compression Protocol"; Informational; Aug. 1996; pp. 1-20.
Schneider, et al.; "PPP for Data Compression in Data Circuit-Terminating Equipment (DCE)"; Informational; Aug. 1996; pp. 1-10.
Atkins, et al.; "PGP Message Exchange Formats"; Informational; Aug. 1996; pp. 1-21.
Castineyra, et al.; "The Nimrod Routing Architecture"; Informational, Aug. 1996; pp. 1-27.
Freed, et al.; "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures"; Best Current Practice; Nov. 1996; pp. 1-21.
Shacham, et al.; "IP Payload Compression Protocol (IPComp)"; Standards Track; Dec. 1998; pp. 1-10.
Sidewinder 50 Product Manual; Seagate Technology, Inc.; 1997.
IBM RAMAC Virtual Array; IBM; Jul. 1997.
Bruni, et al.; "DB2or OS/390 and Data Compression" IBM Corporation; Nov. 1998.
Smith, Mark; "Thin Client/Server Computing Works"; WindowsITPro; Nov. 1, 1998; pp. 1-13; license.icopyright.net/user/downloadLicense.act?lic=3.7009-8355; accessed Dec. 2, 2008.
International Telecommunication Union; "Information Technology—Digital Compression and Coding of Continuous-Tone Still Images-Requirements and Guidelines"; 1993.
International Telecommunications Union; "Information technology — Lossless and near-lossless compression of continuous-tone still images — Baseline"; 1999.
Davis, Andrew W.; "The Video Answering Machine: Intel ProShare's Next Step"; Advanced Imaging; vol. 12, No. 3; Mar. 1997; pp. 28, 30.
Abbott, Ill, Walter D.; "A Simple, Low Overhead Data Compression Algorithm for Converting Lossy Compression Processes to Lossless"; Naval Postgraduate School Thesis; Dec. 1993.
Thomborson, Clark; "V.42bis and Other Ziv-Lempel Variants"; IEEE; 1991; p. 460.
Thomborson, Cork; "The V.42bis Standard for Data-Compressing Modems"; IEEE; Oct. 1992; pp. 41-53.
Sun, Andrew; "Using and Managing PPP"; O'Reilly & Associates, Inc.; 1999.
"What is the V42bis Standard?"; www.faqs.org/faqs/compression-faq/part1/section-10.html; accessed on Dec. 2, 2008.
"The WSDC Download Guide: Drive Image Professional for DOS, OS/2, and Windows"; wsdcds01.watson.ibm.com/WSDC.nsf/Guides/Download/Applications-DriveImage.htm; Accessed Nov. 22, 1999.
"The WSDC Download Guide: Drive Image Professional"; wsdcds01.watson.ibm.com/wsdc.nsf/Guides/Download/Applications-DirveImage.htm; accessed on May 3, 2001.
APPNOTE_TXT from pkware.txt; Version 6.3.2; PKWARE Inc., 1989.
CU-SeeMe readme.txt; Dec. 2, 1995.
CU-seeme txt from indstate.txt; README.TXT for CU-SeeMe version 0.90b1; Mar. 23, 1997.
Cuseeme txt 19960221.txt; CUSEEME.TXT; Feb. 21, 1996.

Citrix Technology Guide. 1997.
Lettieri, et al.; "Data Compression in the V.42bis Modems"; pp. 398-403. 1990.
High Performance x2/V.34+/V.42bis 56K BPS Plug & Play External Voice/FAX/Data Modem User's Manual. 1997.
H.323 Protocols Suite; www.protocols.com/pbook/h323.htm.
Hoffman, Roy; "Data Compression in Digital Systems"; Chapman & Hall; 1997; Chapter 14, pp. 344-360.
LBX X Consortium Algorithms; rzdocs.uni-hohenheim.de/aix_4.33/ext_doc/usr/share/man/info/en_US/a_doc_lib/x 11..";X11R6 Technical Specifications.
Basics of Images; www.geom.uiuc.edu/events/courses/1996/cmwh/Stills/basics.html.
Parties' Joint Claim Construction and Prehearing Statement Pursuant to P.R. 4-3, filed in *Realtime Data, LLC d/b/a/ IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas Feb. 18, 2009.
Declaration of Professor James A. Storer, Ph.D., relating to U.S. Patent No. 6,604,158, Mar. 18, 2009.
Declaration of Professor James A. Storer, Ph.D., relating to U.S. Patent No. 6,601,104, Mar. 18, 2009.
Declaration of Professor James A. Storer, Ph.D., relating to U.S. Patent No. 7,321,937, May 4, 2009.
Declaration of Professor James A. Storer, Ph.D., relating to U.S. Patent No. 6,624,761, May 4, 2009.
Declaration of Professor James A. Storer, Ph.D., relating to U.S. Patent No. 7,378,992, May 20, 2009.
Declaration of Professor James A. Storer, Ph.D., relating to U.S. Patent No. 7,161,506, May 26, 2009.
"Video Coding for Low Bit Rate Communication", International Telecommunication Union (ITU),Recommendation H.263, § 3.4 (Mar. 1996).
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 6,624,761, Control No. 95/000,464, issued Jul. 24, 2009, 29 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 6,624,761, Control No. 95/000,464, issued Dec. 15, 2009, 20 pgs.
Response to Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, Control No. 95/000,466, filed Oct. 13, 2009, 26 pgs.
Response to Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, Control No. 95/000,466, filed Aug. 24, 2009, 39 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, Control No. 95/000,466, issued Jun. 22, 2009, 11 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 7,321,937, Control No. 95/000,466, issued Jun. 22, 2009, 16 pgs.
Official Action Closing Prosecution for Inter Partes Reexamination of U.S. Pat. No. 7,321,937, Control No. 95/000,466, issued Dec. 22, 2009, 12 pgs.
Comments by Third Party Requester to Patent Owner's Response Inter Partes Reexamination of U.S. Patent No. 7,321,937, Control No. 95/000,466, filed Nov. 10, 2009, 30 pgs.
Supplemental Declaration of Professor James A. Storer, Ph.D. under 37 C.F.R. §1.132 in Inter Partes Reexamination of U.S. Patent No. 7,321,937, Control No. 95/000,466, executed on Nov. 10, 2009, 16 pgs.
Examiner Interview Summary in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, Control No. 90/009,428, issued Dec. 3, 2009, 3 pgs.
Response to Office Action in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, Control No. 90/009,428, filed Dec. 28, 2009, 13 pgs.
Non-Final Office Action in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, Control No. 90/009,428, issued Nov. 2, 2009, 13 pgs.
Official Order Granting Request for Ex Parte Reexamination of U.S. Pat. No. 6,601,104, Control No. 90/009,428, issued Jun. 1, 2009, 12 pgs.
Declaration of Dr. George T. Ligler under 37 C.F.R. §1.132 in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, Control No. 90/009,428, executed Dec. 28, 2009 16 pgs.
Supplementary Declaration of Dr. George T. Ligler under 37 C.F.R. §1.132 in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, Control No. 90/009,428, executed Dec. 30, 2009 1 pg.
Declaration of Dr. George T. Ligler under 37 C.F.R. §1.132 in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, Control No. 95/000,466, executed Aug. 24, 2009 16 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 7,161,506, Control No. 95/000,479, issued Aug. 14, 2009, 41 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,161,506, Control No. 95/000,479, issued Dec. 15, 2009, 37 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 7,378,992, Control No. 95/000,478, issued Aug. 13, 2009, 60 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,378,992, Control No. 95/000,478, issued Dec. 15, 2009, 27 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 6,604,158 Control No. 95/000,486, issued Aug. 14, 2009, 35 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 6,604,158, Control No. 95/000,486, issued Nov. 12, 2009, 199 pgs.
Expert Report of Dr. James A. Storer on Invalidity filed on behalf of some of the defendants [Includes Appendices - Exhibits A-K (Exhibit A has been redacted pursuant to a protective order)] filed in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, Jun. 10, 2009, 1090 pgs.
Supplementary Expert Report of Dr. James A. Storer on Invalidity filed on behalf of some of the defendants [Includes Appendices - Exhibits 1-8] filed in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-Led; U.S. District Court for the Eastern District of Texas, Jun. 19, 2009, 301 pgs.
Expert Report of James B. Gambrel on Inequitable Conduct filed on behalf of some of the defendants [Includes Appendices - Exhibits A-I] filed in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-Led; U.S. District Court for the Eastern District of Texas, Jun. 10, 2009, 199 pgs.
Report and Recommendation of United States Magistrate Judge on Motion for Partial Summary Judgment issued on Jun. 23, 2009, in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-Led; U.S. District Court for the Eastern District of Texas, 22 pgs.
Order Adopting Report and Recommendation of United States Magistrate Judge, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, Aug. 24, 2009, 2 pgs.
Opinion and Order of United States Magistrate Judge regarding Claim Construction, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, issued Jun. 22, 2009, 75 pgs.
Script for Defendants' Joint Claim Construction Technology Tutorial Presented to the Magistrate Judge in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, no date on document, 95 pgs.
Script for Realtimes' Technology Tutorial Presented to the Magistrate Judge in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.* Civil Action No. 6:08-cv00144-LED; U.S. District Court for the Eastern District of Texas, Mar. 16, 2009, 69 pgs.
Opinion and Order of United States Magistrate Judge regarding Plaintiffs Motion to Strike Unauthorized New Invalidity Theories from Defendant Citrix's Opening and Reply Briefs in Support of its Motion for Summary Judgment of Invalidity, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, issued Dec. 8, 2009, 10 pgs.
Declaration of Patrick Gogerty, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, executed May 8, 2009, 3 pgs.
Defendant Citrix Systems, Inc.'s Notice Pursuant to 35 U.S.C. Section 282 Disclosures, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 7 pgs.

Blue Coat Defendants' Notice Pursuant to 35 U.S.C. Section 282 Disclosures, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 7 pgs.
Expand Networks' 35 U.S.C. Section 282 Disclosures, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 4 pgs.
Expand Networks' 35 U.S.C. Section 282 Disclosures (Amended), *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 5 pgs.
Defendant Citrix Systems, Inc.'s Notice of Obviousness Combinations Pursuant to Court Order, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 3 pgs.
Order of United States Magistrate Judge regarding Motion to Limit the Number of Prior Art References to be Asserted at Trail, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 21, 2009, 6 pgs.
Expand Defendants' Notice of Obviousness Combinations Pursuant to Court Order, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 22, 2009, 3 pgs.
Blue Coat Systems, Inc. and 7-Eleven, Inc.'s Notice of Obviousness Combinations to be Used at Trial, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 22, 2009, 30 pgs.
Defendant Citrix Systems, Inc's Notice of Other Prior Art References Within the Scope of the References Discussed at the Dec. 17, 2009 Hearing, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 29, 2009, 6 pgs.
Docket Listing downloaded Jan. 4, 2010 for *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Apr. 18, 2008, 161 pgs.
Preliminary Data Sheet, 9600 Data Compressor Processor, Hi/fn, 1997-99, HIFN 000001-68, 68 pgs.
Data Sheet, 9751 Data Compression Processor, 1997-99, HIFN 000069-187, 119 pgs.
Signal Termination Guide, Application Note, Hi/fn, 1997-98, HIFN 000188-194, 7 pgs.
How LZS Data Compression Works, Application Note, Hi/fn, 1997-99, HIFN 000195-207, 13 pgs.
Reference Hardware, 9751 Compression Processor, Hi/fn, 1997-99, HIFN 000208-221, 14 pgs.
Using 9751 in Big Endian Systems, Application Note, Hi/fn, 1997-99, HIFN 000222-234, 13 pgs.
Specification Update, 9751 Compression Processor, Hi/fn, 1997-2000, HIFN 000235-245, 11 pgs.
9732AM Product Release, Hi/fn, 1994-99, HIFN 000246-302, 57 pgs.
Data Sheet, 9732A Data Compression Processor, Hi/fn, 1997-99, HIFN 000303-353, 51 pgs.
9711 to 7711 Migration, Application Note, Hi/fn, 1997-99, HIFN 000354-361, 8 pgs.
Specification Update, 9711 Data Compression Processor, Hi/fn, 1997-99, HFIN 000362-370, 9 pgs.
Differences Between the 9710 & 9711 Processors, Application Note, Hi/fn, 1997-99, HIFN 000371-77, 7 pgs.
Specification Update, 9710 Data Compression Processor, Hi/fn, 1997-99, HIFN 000378-388, 11 pgs.
9706/9706A Data Compression Coprocessor Data Sheet, Stac Electronics, 1991-97, HIFN 000389-473, 85 pgs.
9705/9705A Data Compression Coprocessor, Stac Electronics, 1988-96, HIFN 000474-562, 88 pgs.
9705/9705A Data Compression Coprocessor Data Sheet, Stac Electronics, 1988-96, HIFN 000563-649, 87 pgs.
9700/9701 Compression Coprocessors, Hi/fn, 1997, HIFN 000650-702, 53 pgs.
Data Sheet 9610 Data Compression Processor, Hi/fn, 1997-98, HIFN 000703-744, 42 pgs.
Specification Update 9610 Data Compression Processor, Hi/fn, 1997-99, HIFN 000745-751, 7 pgs.
9705 Data Compression Coprocessor, Stac Electronics, 1988-92, HIFN 000752-831, 80 pgs.
9705 Network Software Design Guide, Application Note, Stac Electronics, 1990-91, HIFN 000832-861, 30 pgs.
Data Sheet 9601 Data Compression Processor, Hi/fn, May 21, 1998, HIFN 000862-920, 59 pgs.
7751 Encryption Processor Reference Kit, Hi/fn, Apr. 1999, HIFN 000921-1114, 194 pgs.
Hardware Data Book, Hi/fn, Nov. 1998, HIFN 001115-1430, 316 pgs.
Data Compression Data Book, Hi/fn, Jan. 1999, HIFN 001431-1889, 459 pgs.
Reference Software 7751 Encryption Processor, Hi/fn, Nov. 1998, HIFN 002164-2201, 38 pgs.
Interface Specification for Synergize Encoding/Decoding Program, JPB, Oct. 10, 1997, HIFN 002215-2216, 2 pgs.
Anderson, Chip, Extended Memory Specification Driver, 1998, HIFN 002217-2264, 48 pgs.
Whiting, Doug, LZS Hardware API, Mar. 12, 1993, HIFN 002265-68, 4 pgs.
Whiting, Doug, Encryption in Sequoia, Apr. 28, 1997, HIFN 002309-2313, 5 pgs.
LZS221-C Version 4 Data Compression Software, Data Sheet, Hi/fn, 1994-97, HIFN 002508-2525, 18 pgs.
eXtended Memory Specification (XMS), ver. 2.0, Microsoft, Jul. 19, 1988, HIFN 002670-2683, 14 pgs.
King, Stanley, Just for Your Info—From Microsoft 2, May 4, 1992,HIFN 002684- 2710, 27 pgs.
eXtended Memory Specification (XMS), ver. 2.0, Microsoft, Jul. 19, 1988, HIFN 002711-2724, 14 pgs.
Advanced LZS Technology (ALZS), Whitepaper, Hi/fn, Jun. 1, 1998, HIFN 002725-2727, 3 pgs.
Secure Tape Technology (STT) Whitepaper, Hi/fn, Jun. 1, 1998, HIFN 002728-2733, 6 pgs.
SSLRef 3.0 Api Details, Netscape, Nov. 19, 1996, HIFN 002734-2778, 45 pgs.
LZS221-C Version 4 Data Compression Software Data Sheet , Hi/fn, 1994-97, HIFN 002779-2796, 18 pgs.
MPPC-C Version 4 Data Compression Software Data Sheet, Hi/fn, 1994-1997, HIFN 002797-2810, 14 pgs.
Magstar MP Hardware Reference B Series Models Document GA32-0365-01, 1996-1997, [IBM_1_601 pp. 1-338], 338 pages.
Magstar MP 3570 Tape Subsystem, Operator Guide, B-Series Models, 1998- 1999, [IBM_1_601 pp. 339-525], 187 pages.
Preview, IBM Magstar 3590 Tape System Enhancements, Hardware Announcement, Feb. 16, 1999, [IBM_1_601 pp. 526-527], 2 pgs.
New IBM Magstar 3590 Models E11 and E1A Enhance Tape Drive Performance, Hardware Announcement, Apr. 20, 1999, [IBM_1_601 pp. 528-540] 13 pgs.
New IBM Magstar 3590 Model A60 Dramatically Enhances Tape Drive Performance, Hardware Announcement Jul. 27, 1999, [IBM_1_601 pp. 541550] 10 pgs.
The IBM Magstar Mp Tape Subsystem Provides Fast Access to Data, Sep. 3, 1996, Announcement No. 196-176, [IBM_1_601 pp. 551-563] 13 pgs.
IBM 3590 High Performance Tape Subsystem, Apr. 10, 1995, Announcement 195-106, [IBM_1_601 pp. 564-581] 18 pgs.
Standard ECMA-222 (Jun. 1995): ECMA — Standardizing Information and Communications Systems, Adaptive Lossless Data Compression Algorithm, [IBM_1_601 pp. 564-601] 38 pgs.
IBM 3590 and 3494 Revised Availability, Hardware Announcement Aug. 8, 1995, [IBM_743_1241 p. 1] 1 pg.
Direct Delivery of IBM 3494, 3466, and 3590 Storage Products, Hardware Announcement, Sep. 30, 1997, Announcement 197-297, [IBM_743_1241 pp. 2-3] 2 pgs.
IBM Magstar 3590 Enhances Open Systems, Hardware Announcement Feb. 9, 1996, Announcement 198-014, [IBM_743_1241 pp. 4-7] 4 pgs.
Hardware Withdrawal: IBM Magstar 3590 A00 Controller — Replacement Available, Announcement No. 197-267, Withdrawal Announcement, Dec. 9, 1997, [IBM_743_1241 p. 9] 1 pg.

IBM Magstar 3590 Tape Subsystem, Introduction and Planning Guide, Document No. GA32-0329007, [IBM_743_1241 pp. 10-499] 490 pgs.
NetMeeting 2.0 Reviewers Guide, Apr. 1997, [MSCS_298_339] 42 pgs.
Microsoft NetMeeting Compatible Products and Services Directory, Apr. 1997, [MSCS_242297] 56 pgs.
Microsoft NetMeeting "Try This!" Guide, 1997, [MSCS_340_345] 6 pgs.
The Professional Companion to NetMeeting 2 — the Technical Guide to Installing, Configuring, and Supporting NetMeeting 2.0 in Your Organization -Microsoft NetMeeting 2.0, 1996-97, [MSCS_2_241] 240 pgs.
CUSeeMe 3.1.2 User Guide, Nov. 1998, [RAD_1_220] 220 pgs.
MeetingPoint Conference Server Users Guide 3.0, Nov. 1997, [RAD_221_548] 328 pgs.
MeetingPoint Conference Server Users Guide 4.0.2, Dec. 1999, [RAD_549_818] 270 pgs.
MeetingPoint Conference Service Users Guide 3.5.1, Dec. 1998, [RAD_819_1062] 244 pgs.
Enhanced CUSeeMe — Authorized Guide, 1995-1996, [RAD_1063_1372] 310 pgs.
Meeting Point Reader File, Jun. 1999, [RAD_1437_1445] 9 pgs.
Press Release - White Pine Announces Launch of MeetingPoint Conferences Server, Oct. 9, 1997, [RAD_1738_1739] 2 pgs.
Press Release - Leading Network Service Providers Line Up to Support White Pine's Meeting Point Conference Server Technology, Oct. 9, 1997, [RAD_1740_1743] 4 pgs.
BYTE—A New MeetingPoint for Videoconferencing, Oct. 9, 1997, [RAD_1744_1750] 7 pgs.
Notice of Allowance in Commonly-Assigned U.S. Appl. No. 11/651,366, issued Apr. 10, 2009, 7 pgs.
Amendment under 37 C.F.R. §1.132 in Commonly-Assigned U.S. Appl. No. 11/651,366, filed Jul. 30, 2008, 18 pgs.
CCITT Draft Recommendation T.4, RFC 804, Jan. 1981, 12 pgs.
SNA Formats, IBM Corporation, 14^{th} Ed., Nov. 1993, 3 pgs.
Munteanu et al, "Wavelet-Based Lossless Compression Scheme with Progressive Transmission Capability," John Wiley & Sons, Inc., Int'l J. Imaging Sys. Tech., vol. 10, (1999) pp. 76-85.
Forchhammer and Jensen, "Data Compression of Scanned Halftone Images," IEEE Trans. Commun., vol. 42, Feb.-Apr. 1994, pp. 1881-1893.

* cited by examiner

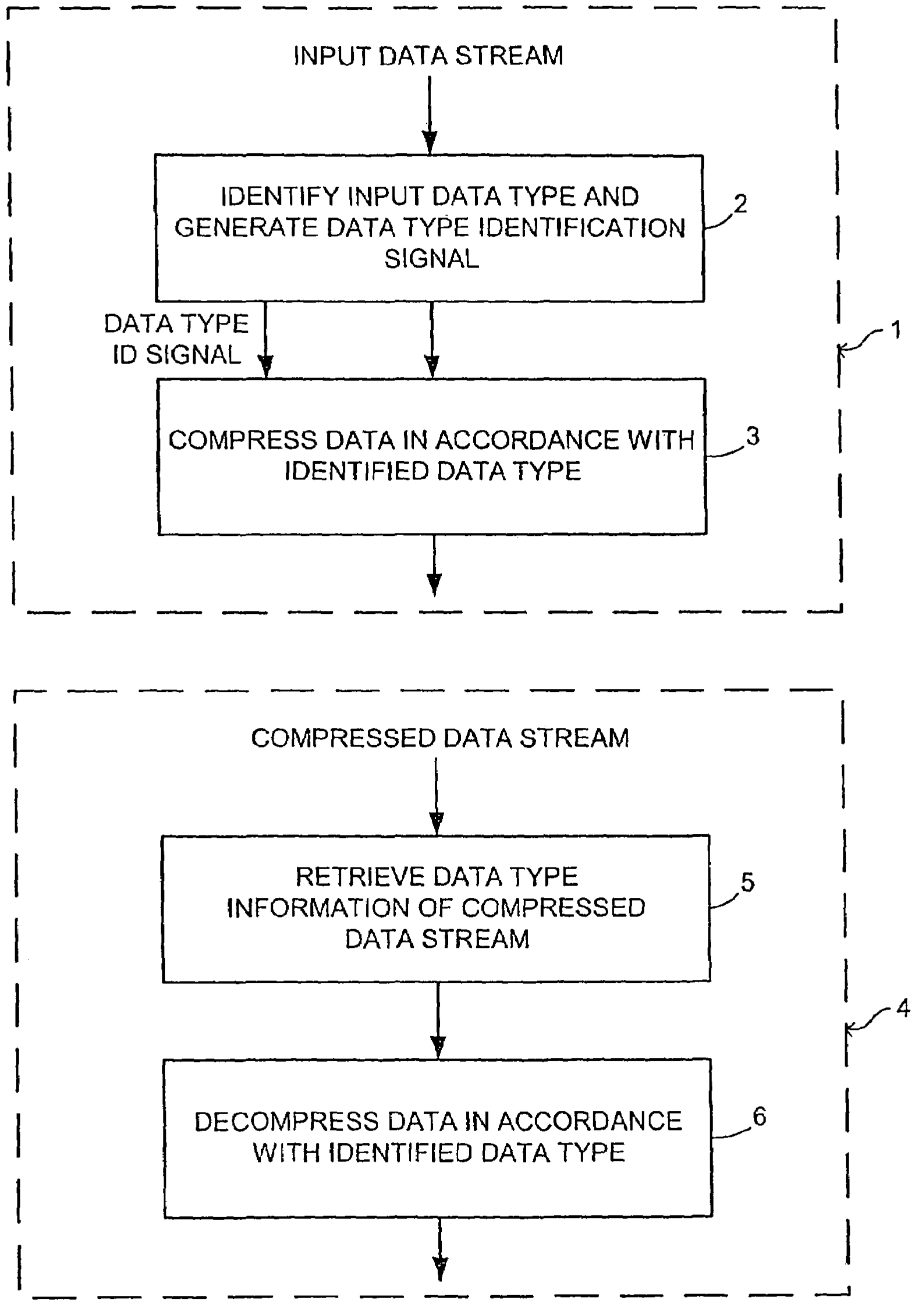


FIG. 1
PRIOR ART

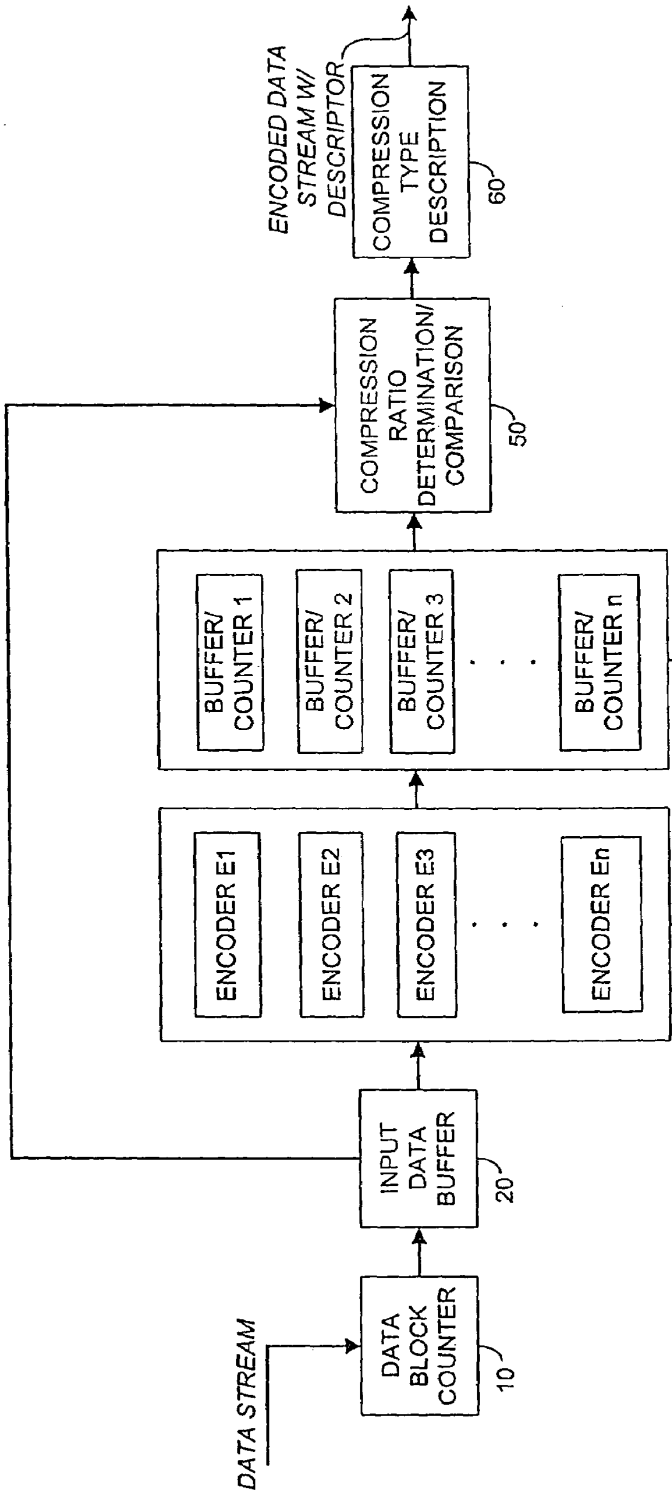
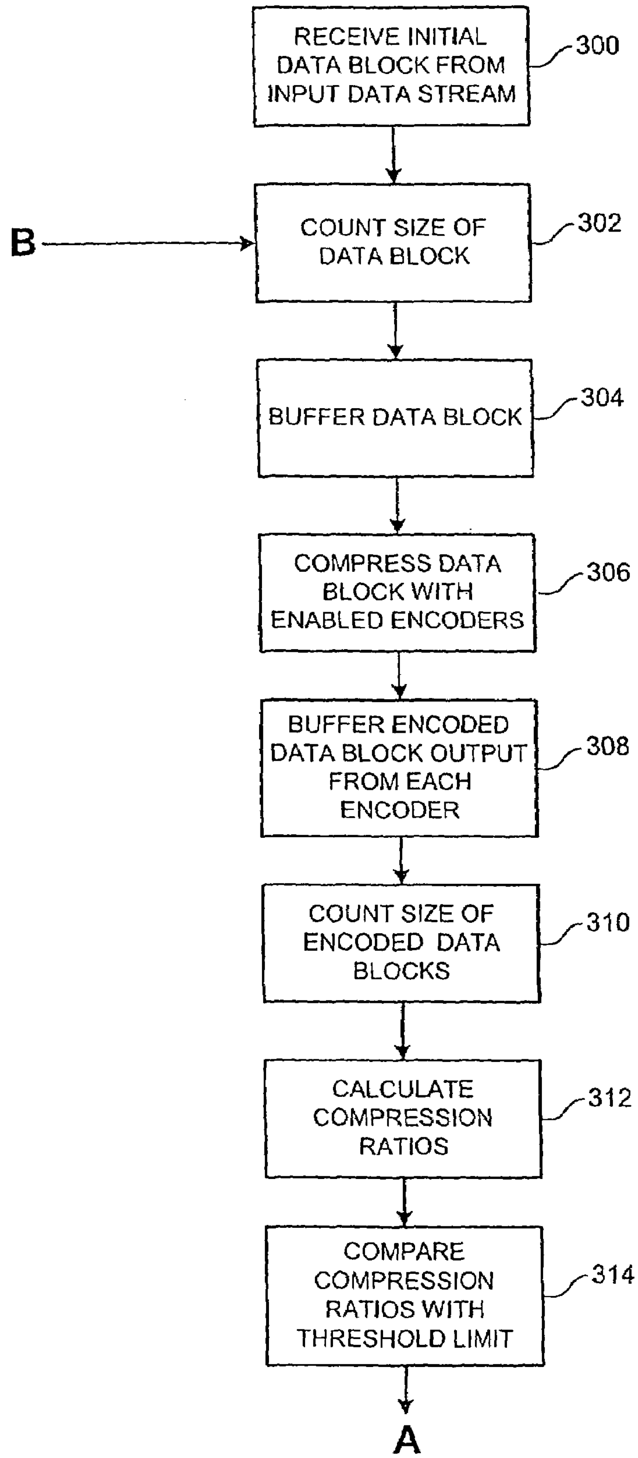

DATA STREAM
DATA BLOCK COUNTER
10
INPUT DATA BUFFER
20
ENCODER E1
ENCODER E2
ENCODER E3
ENCODER En
30
BUFFER/ COUNTER 1
BUFFER/ COUNTER 2
BUFFER/ COUNTER 3
BUFFER/ COUNTER n
40
COMPRESSION RATIO DETERMINATION/ COMPARISON
50
COMPRESSION TYPE DESCRIPTION
60
ENCODED DATA STREAM W/ DESCRIPTOR

FIG. 2



RECEIVE INITIAL DATA BLOCK FROM INPUT DATA STREAM
300
B
COUNT SIZE OF DATA BLOCK
302
BUFFER DATA BLOCK
304
COMPRESS DATA BLOCK WITH ENABLED ENCODERS
306
BUFFER ENCODED DATA BLOCK OUTPUT FROM EACH ENCODER
308
COUNT SIZE OF ENCODED DATA BLOCKS
310
CALCULATE COMPRESSION RATIOS
312
COMPARE COMPRESSION RATIOS WITH THRESHOLD LIMIT
314
A

FIG. 3a

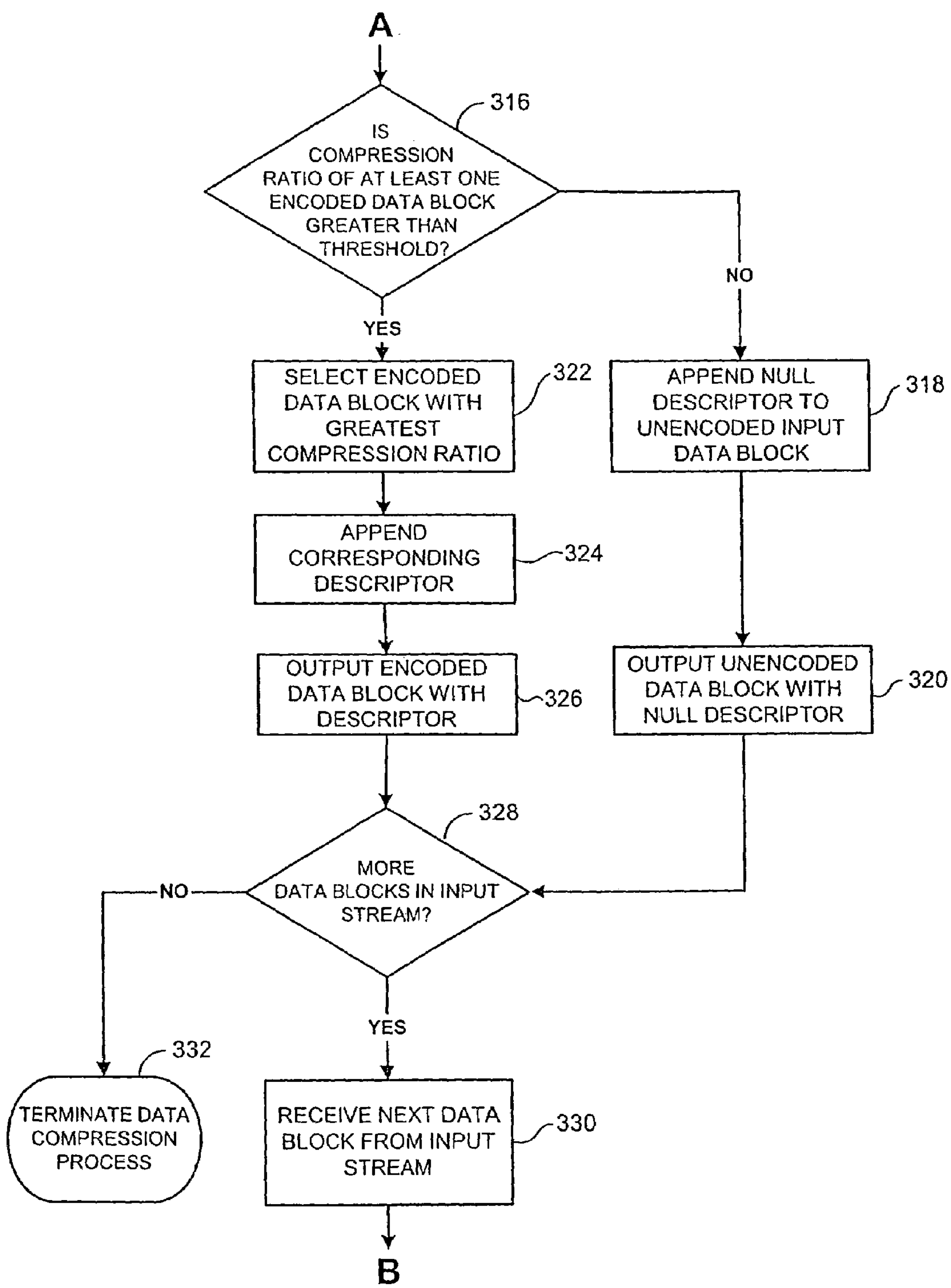

A
IS COMPRESSION RATIO OF AT LEAST ONE ENCODED DATA BLOCK GREATER THAN THRESHOLD?
316
NO
YES
SELECT ENCODED DATA BLOCK WITH GREATEST COMPRESSION RATIO
322
APPEND NULL DESCRIPTOR TO UNENCODED INPUT DATA BLOCK
318
APPEND CORRESPONDING DESCRIPTOR
324
OUTPUT ENCODED DATA BLOCK WITH DESCRIPTOR
326
OUTPUT UNENCODED DATA BLOCK WITH NULL DESCRIPTOR
320
MORE DATA BLOCKS IN INPUT STREAM?
328
NO
YES
TERMINATE DATA COMPRESSION PROCESS
332
RECEIVE NEXT DATA BLOCK FROM INPUT STREAM
330
B

FIG. 3b

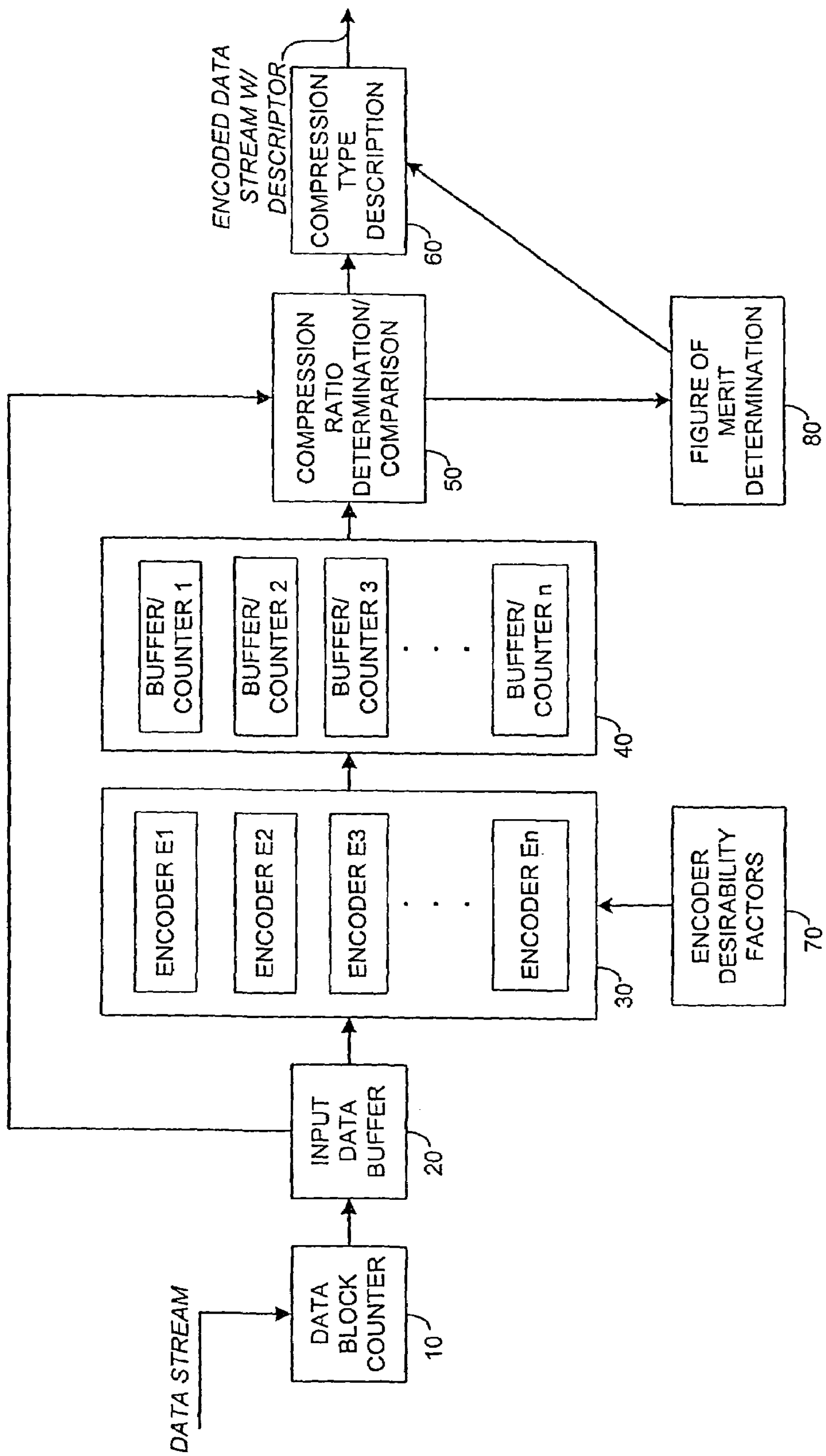

DATA STREAM
DATA BLOCK COUNTER
10
INPUT DATA BUFFER
20
ENCODER E1
ENCODER E2
ENCODER E3
. . .
ENCODER En
30
ENCODER DESIRABILITY FACTORS
70
BUFFER/ COUNTER 1
BUFFER/ COUNTER 2
BUFFER/ COUNTER 3
. . .
BUFFER/ COUNTER n
40
COMPRESSION RATIO DETERMINATION/ COMPARISON
50
FIGURE OF MERIT DETERMINATION
80
COMPRESSION TYPE DESCRIPTION
60
ENCODED DATA STREAM W/ DESCRIPTOR

FIG. 4

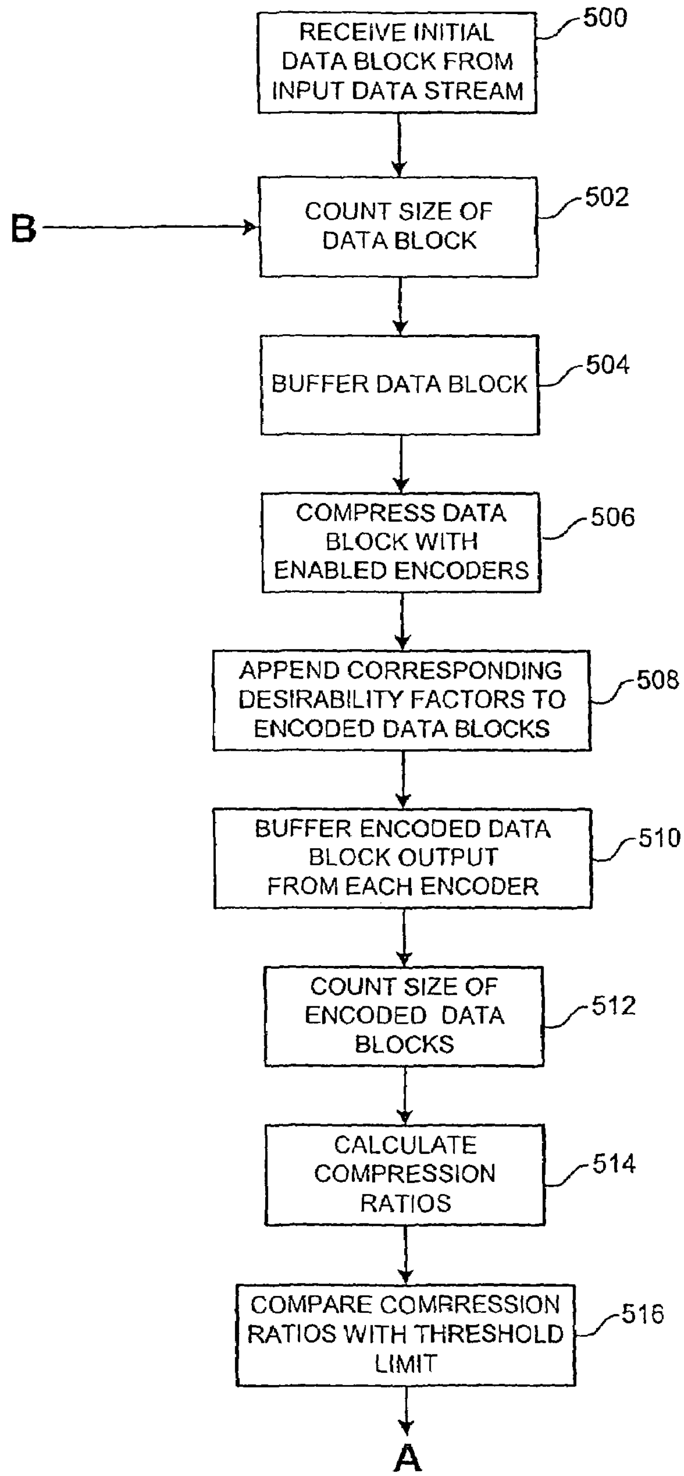

RECEIVE INITIAL DATA BLOCK FROM INPUT DATA STREAM
500
B
COUNT SIZE OF DATA BLOCK
502
BUFFER DATA BLOCK
504
COMPRESS DATA BLOCK WITH ENABLED ENCODERS
506
APPEND CORRESPONDING DESIRABILITY FACTORS TO ENCODED DATA BLOCKS
508
BUFFER ENCODED DATA BLOCK OUTPUT FROM EACH ENCODER
510
COUNT SIZE OF ENCODED DATA BLOCKS
512
CALCULATE COMPRESSION RATIOS
514
COMPARE COMRRESSION RATIOS WITH THRESHOLD LIMIT
516
A

FIG. 5a

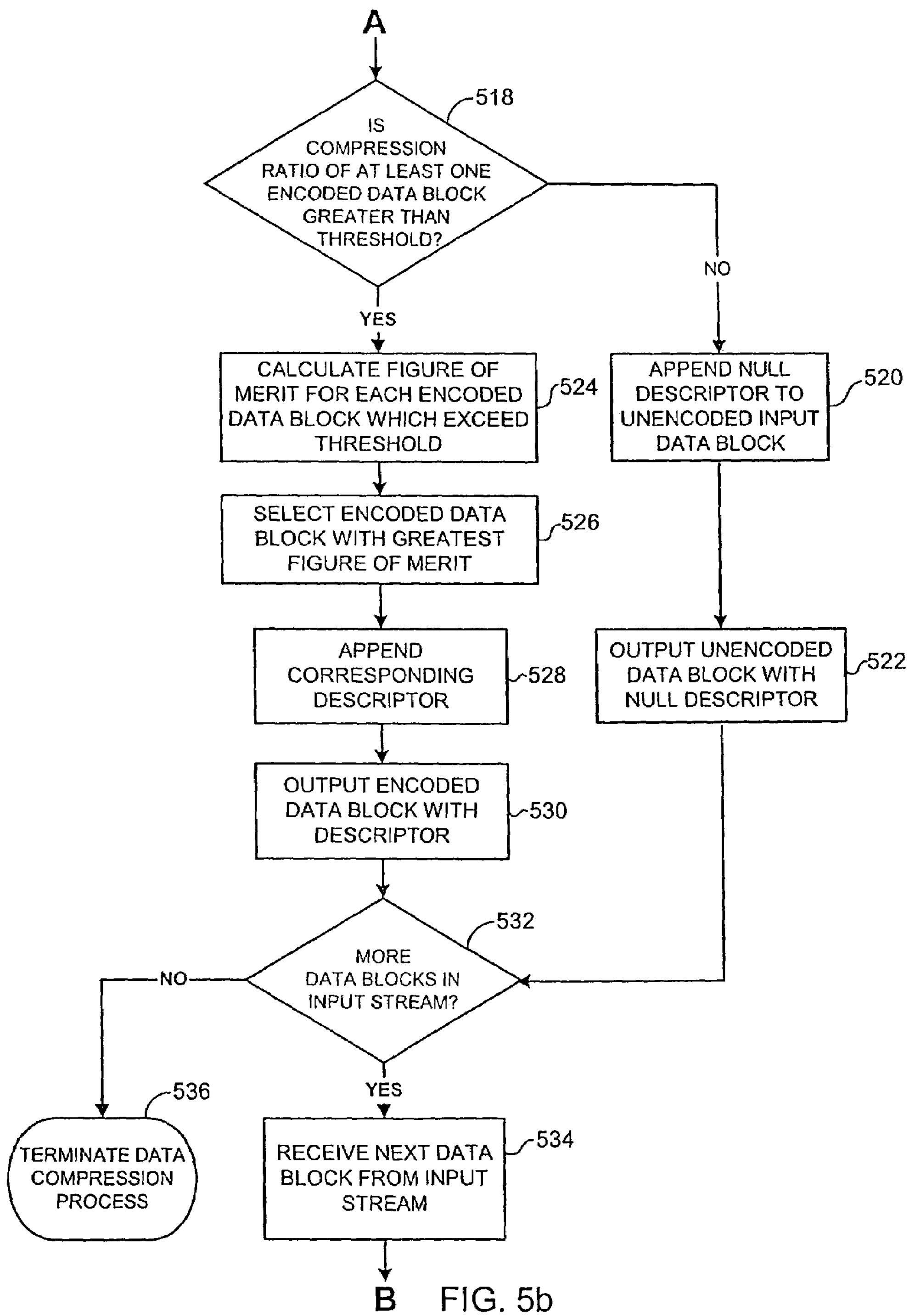

A
IS COMPRESSION RATIO OF AT LEAST ONE ENCODED DATA BLOCK GREATER THAN THRESHOLD?
518
NO
YES
CALCULATE FIGURE OF MERIT FOR EACH ENCODED DATA BLOCK WHICH EXCEED THRESHOLD
524
APPEND NULL DESCRIPTOR TO UNENCODED INPUT DATA BLOCK
520
SELECT ENCODED DATA BLOCK WITH GREATEST FIGURE OF MERIT
526
APPEND CORRESPONDING DESCRIPTOR
528
OUTPUT UNENCODED DATA BLOCK WITH NULL DESCRIPTOR
522
OUTPUT ENCODED DATA BLOCK WITH DESCRIPTOR
530
MORE DATA BLOCKS IN INPUT STREAM?
532
NO
YES
TERMINATE DATA COMPRESSION PROCESS
536
RECEIVE NEXT DATA BLOCK FROM INPUT STREAM
534
B

FIG. 5b

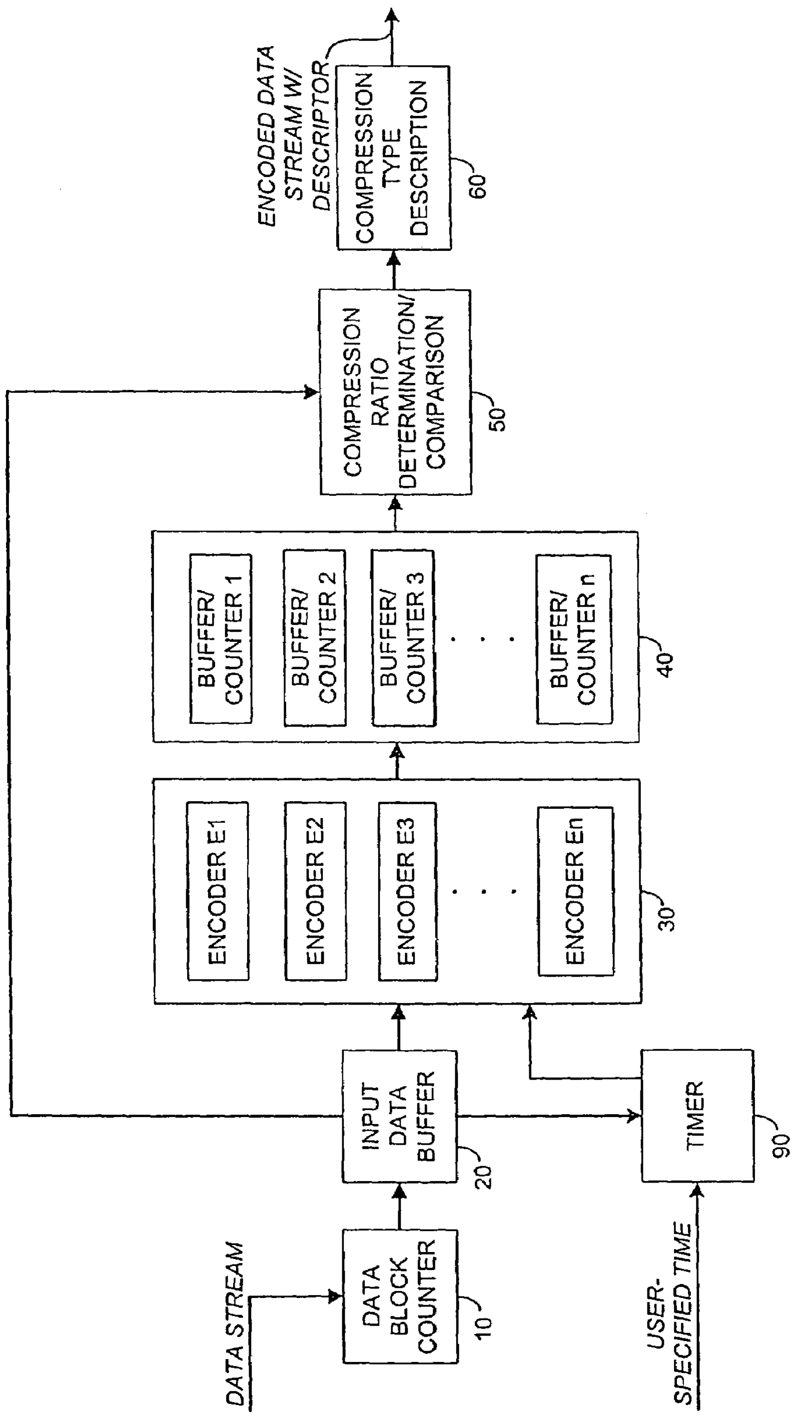
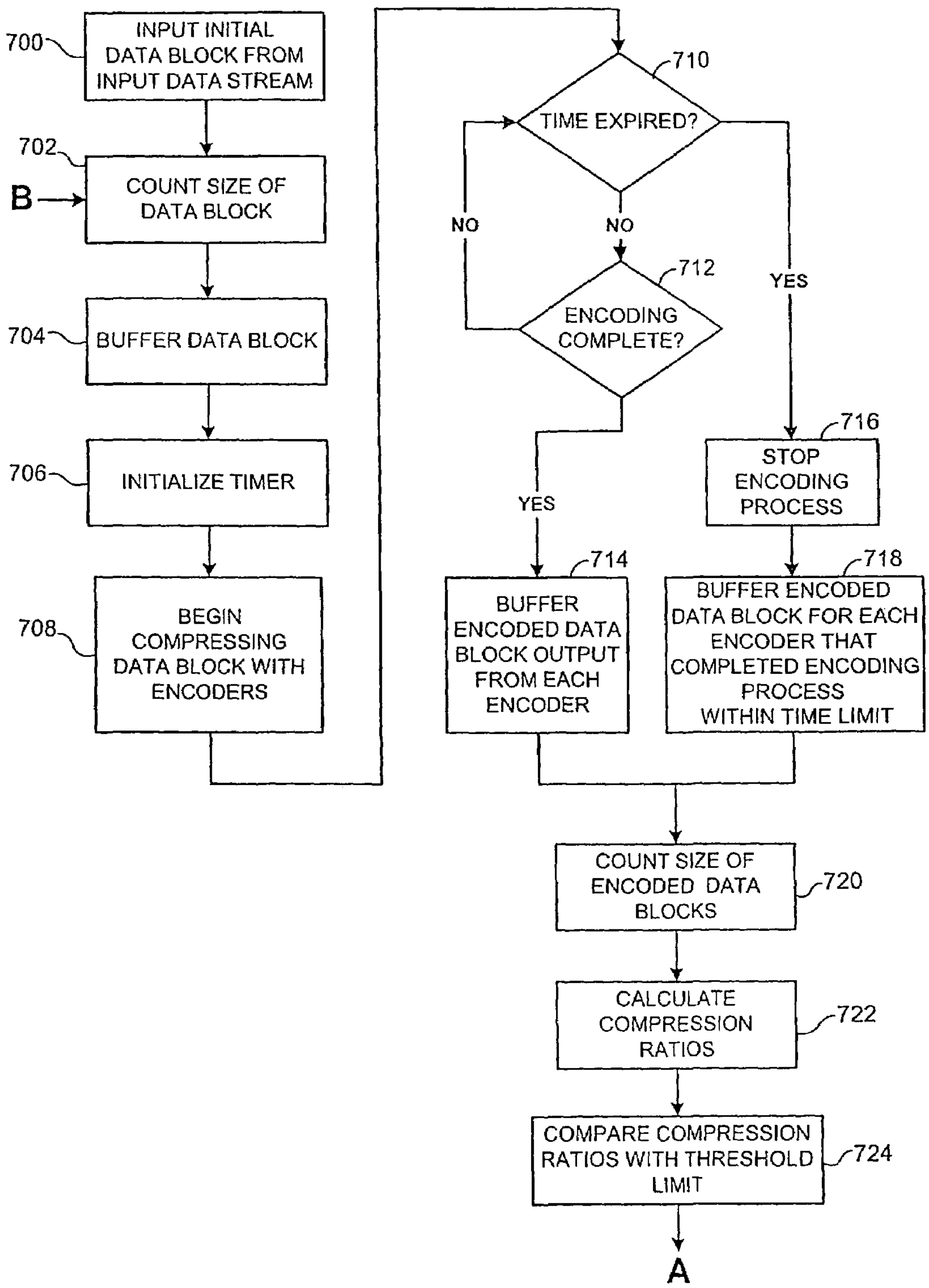

DATA STREAM
DATA BLOCK COUNTER
10
USER-SPECIFIED TIME
INPUT DATA BUFFER
20
TIMER
90
ENCODER E1
ENCODER E2
ENCODER E3
. . .
ENCODER En
30
BUFFER/ COUNTER 1
BUFFER/ COUNTER 2
BUFFER/ COUNTER 3
. . .
BUFFER/ COUNTER n
40
COMPRESSION RATIO DETERMINATION/ COMPARISON
50
COMPRESSION TYPE DESCRIPTION
60
ENCODED DATA STREAM W/ DESCRIPTOR

FIG. 6



700
INPUT INITIAL DATA BLOCK FROM INPUT DATA STREAM
702
B
COUNT SIZE OF DATA BLOCK
704
BUFFER DATA BLOCK
706
INITIALIZE TIMER
708
BEGIN COMPRESSING DATA BLOCK WITH ENCODERS
710
TIME EXPIRED?
NO
NO
712
ENCODING COMPLETE?
YES
YES
716
STOP ENCODING PROCESS
714
BUFFER ENCODED DATA BLOCK OUTPUT FROM EACH ENCODER
718
BUFFER ENCODED DATA BLOCK FOR EACH ENCODER THAT COMPLETED ENCODING PROCESS WITHIN TIME LIMIT
COUNT SIZE OF ENCODED DATA BLOCKS
720
CALCULATE COMPRESSION RATIOS
722
COMPARE COMPRESSION RATIOS WITH THRESHOLD LIMIT
724
A

FIG. 7a

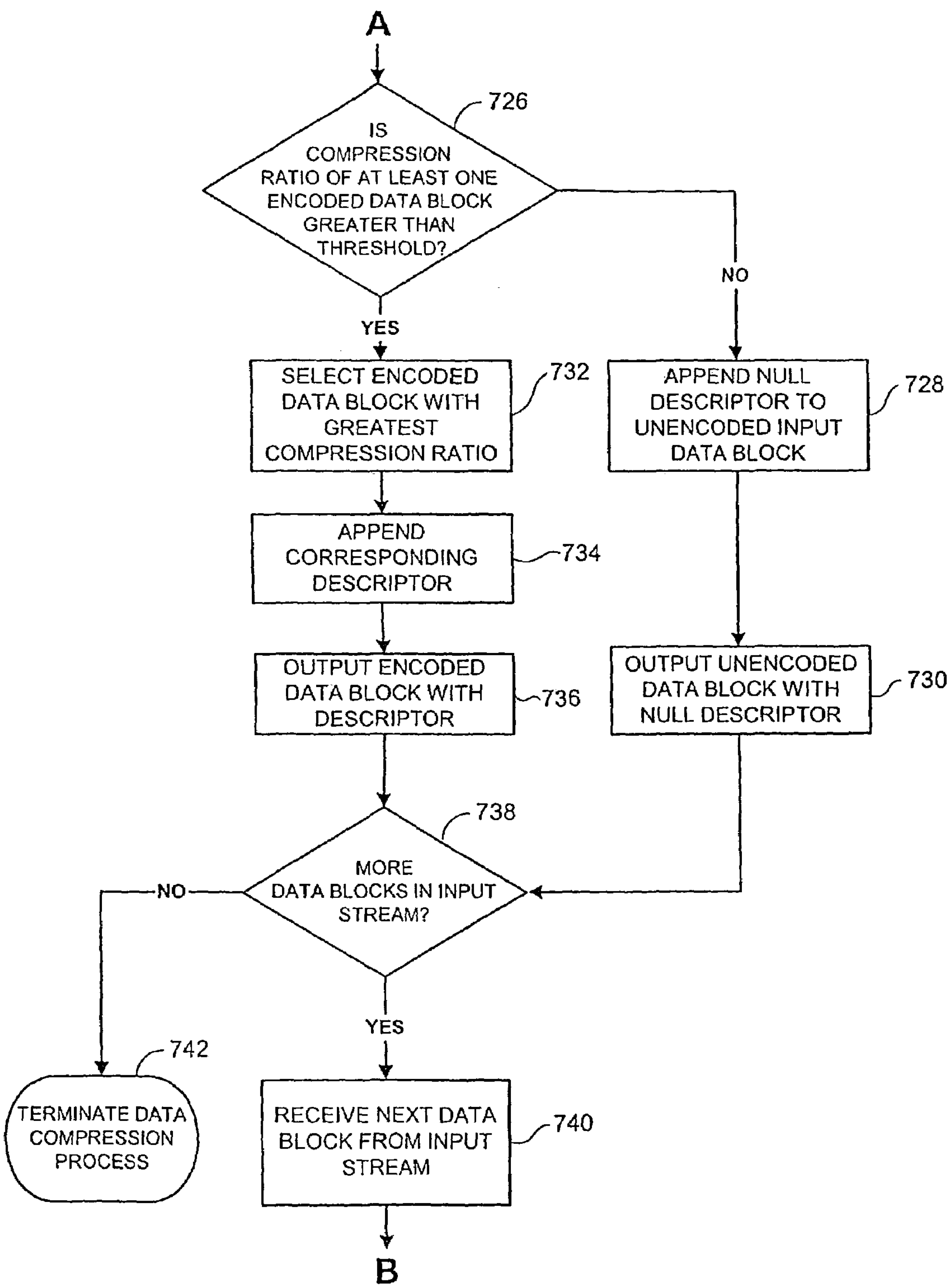

A
IS COMPRESSION RATIO OF AT LEAST ONE ENCODED DATA BLOCK GREATER THAN THRESHOLD?
726
NO
YES
SELECT ENCODED DATA BLOCK WITH GREATEST COMPRESSION RATIO
732
APPEND NULL DESCRIPTOR TO UNENCODED INPUT DATA BLOCK
728
APPEND CORRESPONDING DESCRIPTOR
734
OUTPUT ENCODED DATA BLOCK WITH DESCRIPTOR
736
OUTPUT UNENCODED DATA BLOCK WITH NULL DESCRIPTOR
730
MORE DATA BLOCKS IN INPUT STREAM?
738
NO
YES
742
TERMINATE DATA COMPRESSION PROCESS
RECEIVE NEXT DATA BLOCK FROM INPUT STREAM
740
B

FIG. 7b

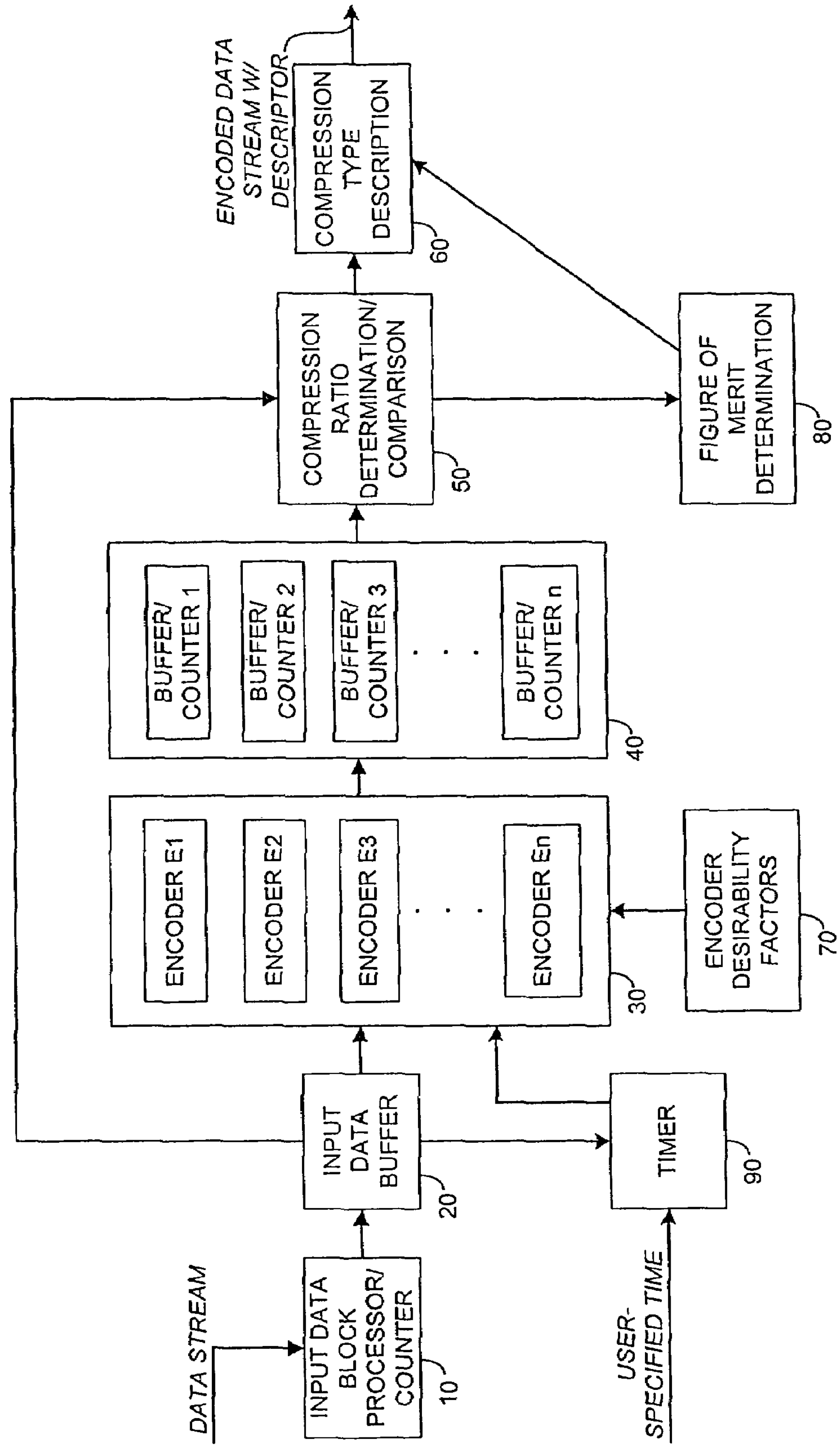

DATA STREAM
INPUT DATA BLOCK PROCESSOR/ COUNTER
10
INPUT DATA BUFFER
20
USER-SPECIFIED TIME
TIMER
90
ENCODER E1
ENCODER E2
ENCODER E3
. . .
ENCODER En
30
ENCODER DESIRABILITY FACTORS
70
BUFFER/ COUNTER 1
BUFFER/ COUNTER 2
BUFFER/ COUNTER 3
. . .
BUFFER/ COUNTER n
40
COMPRESSION RATIO DETERMINATION/ COMPARISON
50
FIGURE OF MERIT DETERMINATION
80
COMPRESSION TYPE DESCRIPTION
60
ENCODED DATA STREAM W/ DESCRIPTOR

FIG. 8

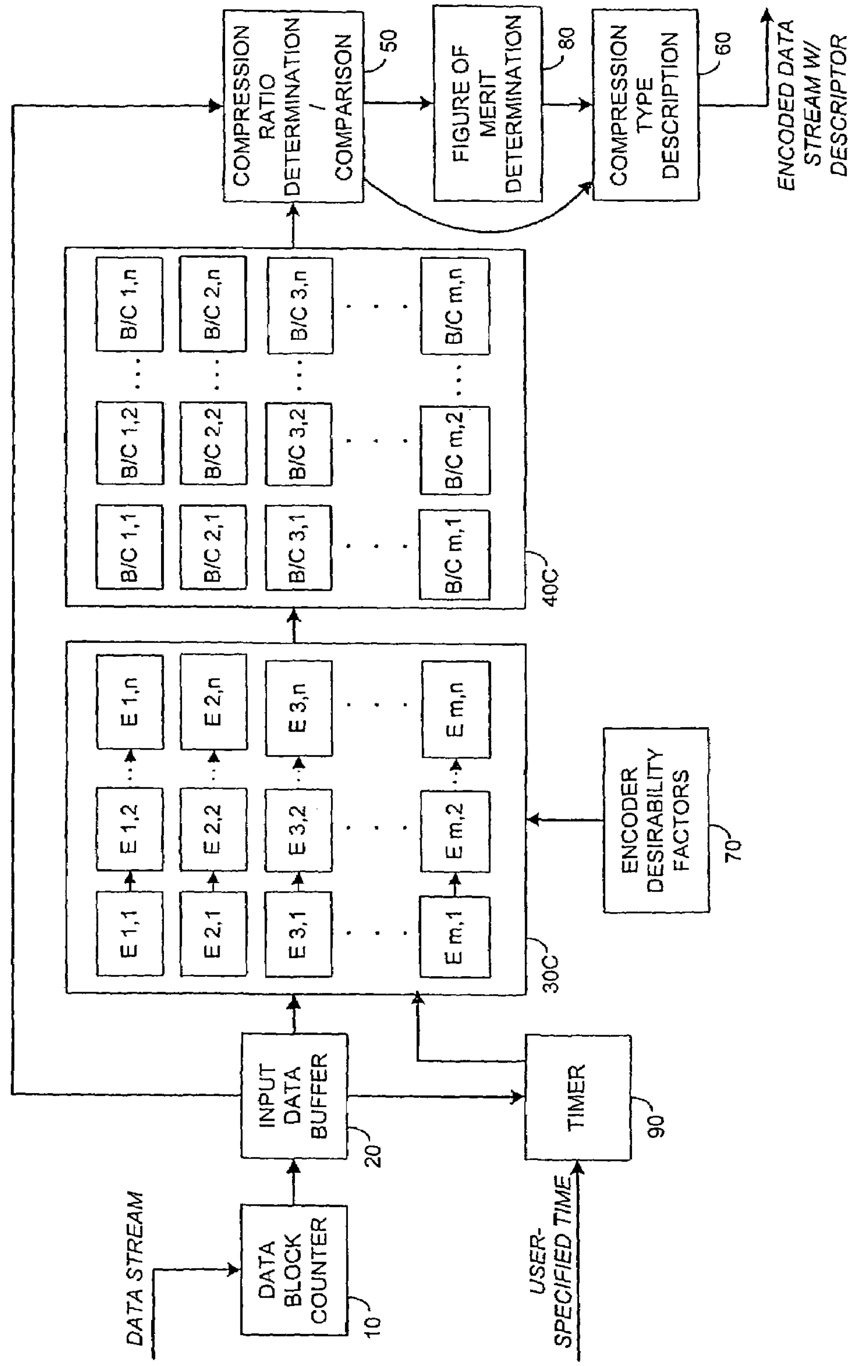
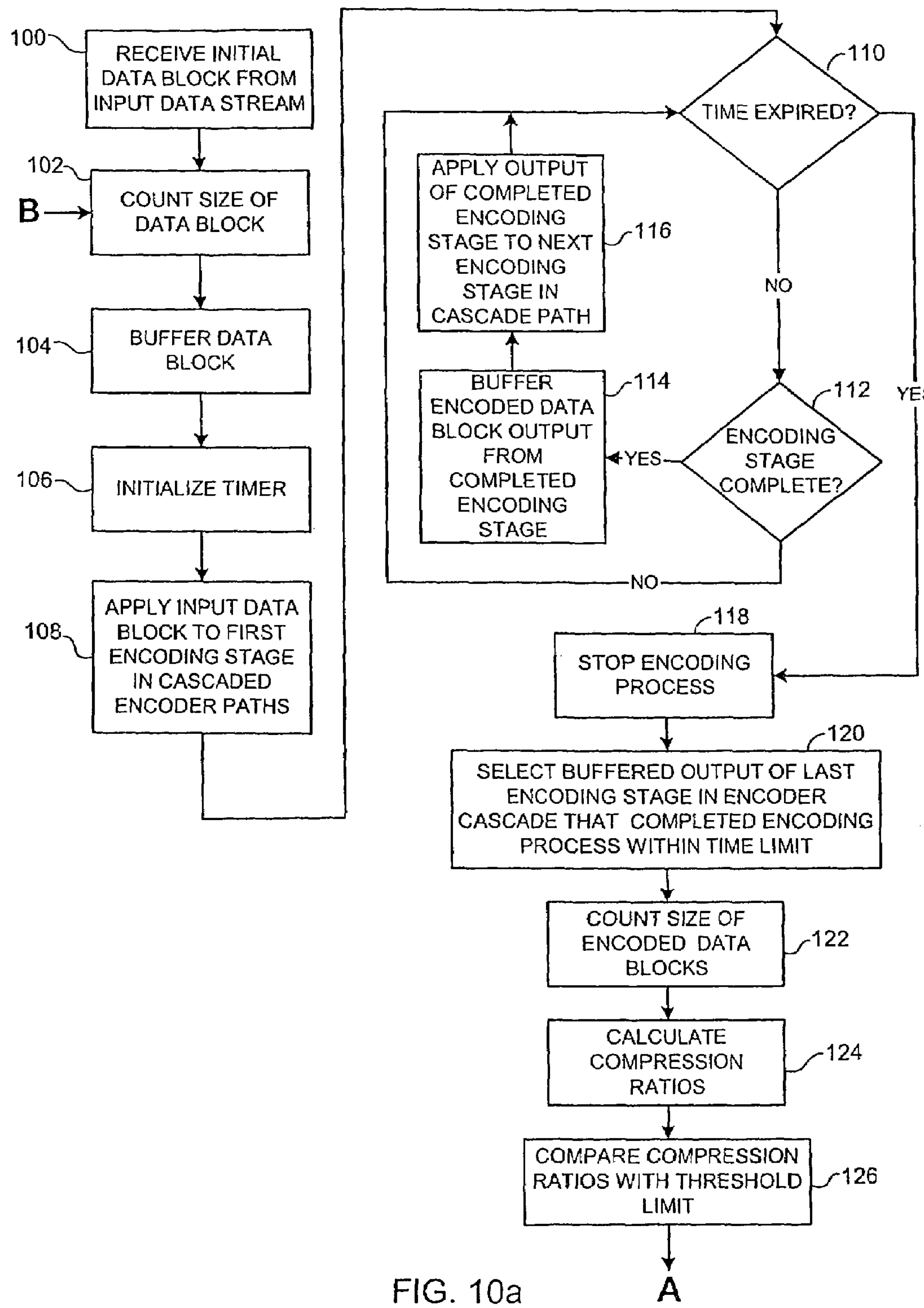

DATA STREAM
DATA BLOCK COUNTER
10
INPUT DATA BUFFER
20
USER-SPECIFIED TIME
TIMER
90
E 1,1
E 1,2
E 1,n
E 2,1
E 2,2
E 2,n
E 3,1
E 3,2
E 3,n
E m,1
E m,2
E m,n
30C
ENCODER DESIRABILITY FACTORS
70
B/C 1,1
B/C 1,2
B/C 1,n
B/C 2,1
B/C 2,2
B/C 2,n
B/C 3,1
B/C 3,2
B/C 3,n
B/C m,1
B/C m,2
B/C m,n
40C
COMPRESSION RATIO DETERMINATION / COMPARISON
50
FIGURE OF MERIT DETERMINATION
80
COMPRESSION TYPE DESCRIPTION
60
ENCODED DATA STREAM W/ DESCRIPTOR

FIG. 9



100
RECEIVE INITIAL DATA BLOCK FROM INPUT DATA STREAM
102
B
COUNT SIZE OF DATA BLOCK
104
BUFFER DATA BLOCK
106
INITIALIZE TIMER
108
APPLY INPUT DATA BLOCK TO FIRST ENCODING STAGE IN CASCADED ENCODER PATHS
110
TIME EXPIRED?
NO
YES
112
ENCODING STAGE COMPLETE?
YES
NO
114
BUFFER ENCODED DATA BLOCK OUTPUT FROM COMPLETED ENCODING STAGE
116
APPLY OUTPUT OF COMPLETED ENCODING STAGE TO NEXT ENCODING STAGE IN CASCADE PATH
118
STOP ENCODING PROCESS
120
SELECT BUFFERED OUTPUT OF LAST ENCODING STAGE IN ENCODER CASCADE THAT COMPLETED ENCODING PROCESS WITHIN TIME LIMIT
122
COUNT SIZE OF ENCODED DATA BLOCKS
124
CALCULATE COMPRESSION RATIOS
126
COMPARE COMPRESSION RATIOS WITH THRESHOLD LIMIT
A

FIG. 10a

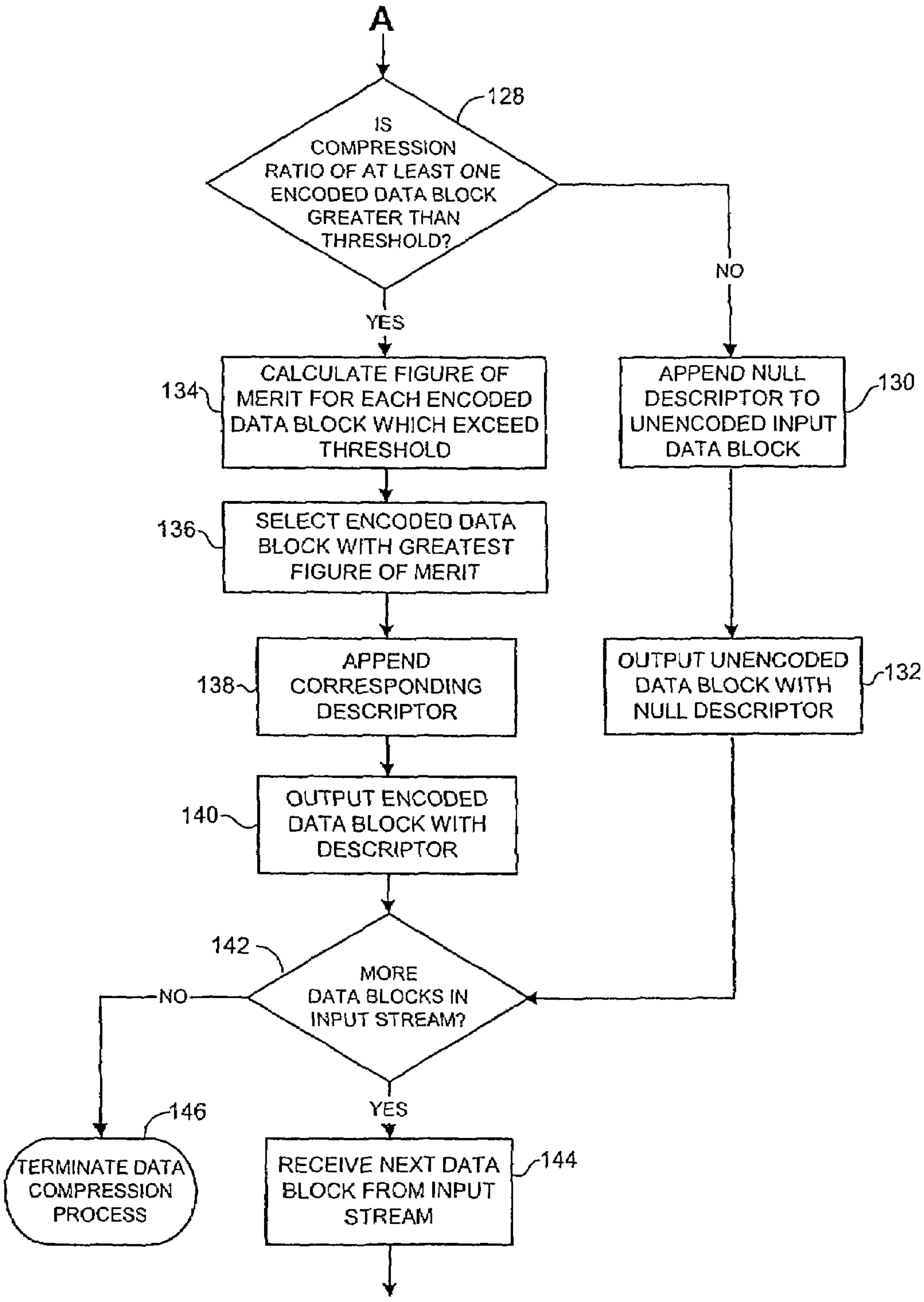

A
IS COMPRESSION RATIO OF AT LEAST ONE ENCODED DATA BLOCK GREATER THAN THRESHOLD?
128
YES
NO
CALCULATE FIGURE OF MERIT FOR EACH ENCODED DATA BLOCK WHICH EXCEED THRESHOLD
134
APPEND NULL DESCRIPTOR TO UNENCODED INPUT DATA BLOCK
130
SELECT ENCODED DATA BLOCK WITH GREATEST FIGURE OF MERIT
136
APPEND CORRESPONDING DESCRIPTOR
138
OUTPUT UNENCODED DATA BLOCK WITH NULL DESCRIPTOR
132
OUTPUT ENCODED DATA BLOCK WITH DESCRIPTOR
140
MORE DATA BLOCKS IN INPUT STREAM?
142
NO
YES
TERMINATE DATA COMPRESSION PROCESS
146
RECEIVE NEXT DATA BLOCK FROM INPUT STREAM
144
B

FIG. 10b

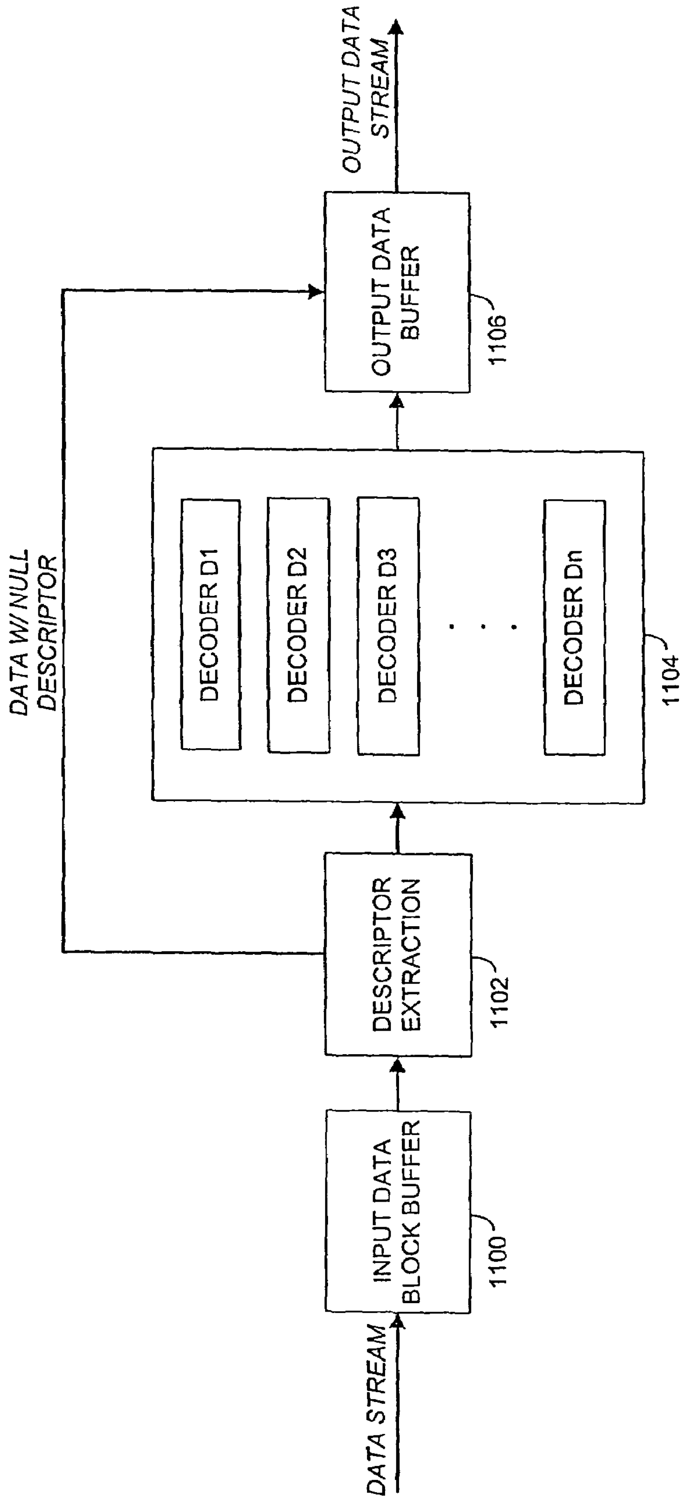

DATA STREAM
INPUT DATA BLOCK BUFFER
1100
DESCRIPTOR EXTRACTION
1102
DATA W/ NULL DESCRIPTOR
DECODER D1
DECODER D2
DECODER D3
DECODER Dn
1104
OUTPUT DATA BUFFER
1106
OUTPUT DATA STREAM

FIG. 11

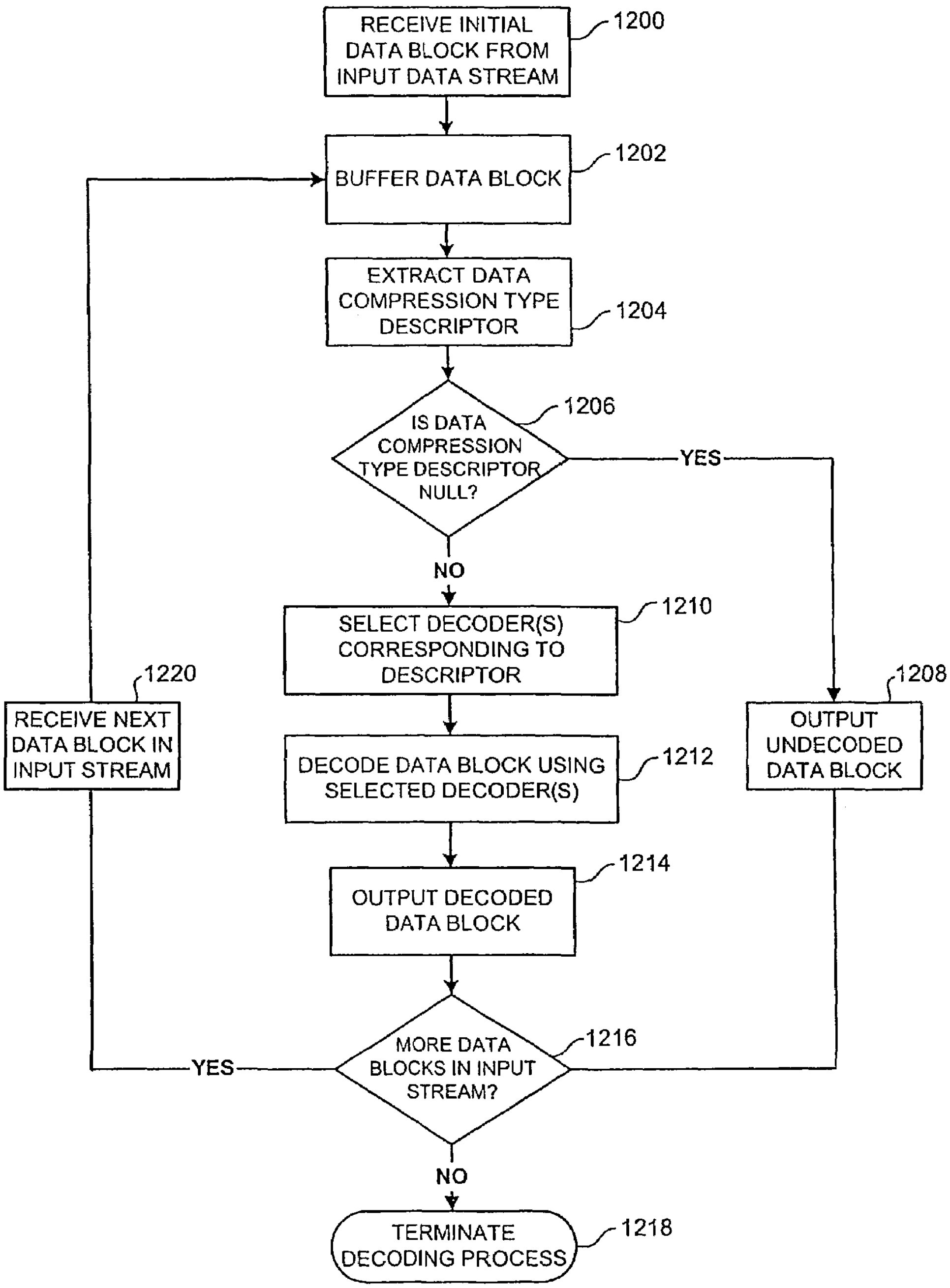

RECEIVE INITIAL DATA BLOCK FROM INPUT DATA STREAM
1200
BUFFER DATA BLOCK
1202
EXTRACT DATA COMPRESSION TYPE DESCRIPTOR
1204
IS DATA COMPRESSION TYPE DESCRIPTOR NULL?
1206
YES
NO
SELECT DECODER(S) CORRESPONDING TO DESCRIPTOR
1210
OUTPUT UNDECODED DATA BLOCK
1208
RECEIVE NEXT DATA BLOCK IN INPUT STREAM
1220
DECODE DATA BLOCK USING SELECTED DECODER(S)
1212
OUTPUT DECODED DATA BLOCK
1214
MORE DATA BLOCKS IN INPUT STREAM?
1216
YES
NO
TERMINATE DECODING PROCESS
1218

FIG. 12

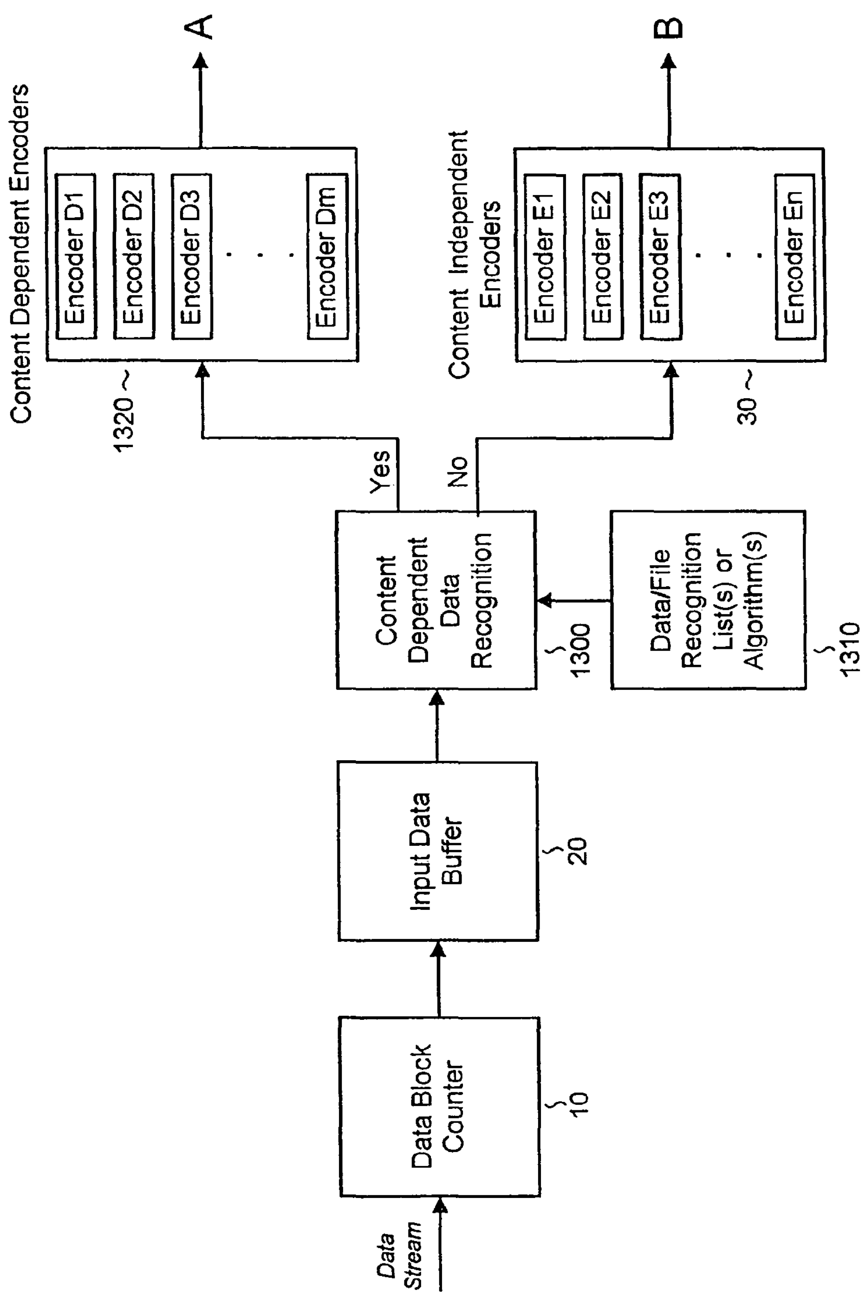

Data Stream
Data Block Counter
10
Input Data Buffer
20
Content Dependent Data Recognition
1300
Data/File Recognition List(s) or Algorithm(s)
1310
Yes
No
Content Dependent Encoders
1320
Encoder D1
Encoder D2
Encoder D3
Encoder Dm
Content Independent Encoders
30
Encoder E1
Encoder E2
Encoder E3
Encoder En
A
B

FIGURE 13A

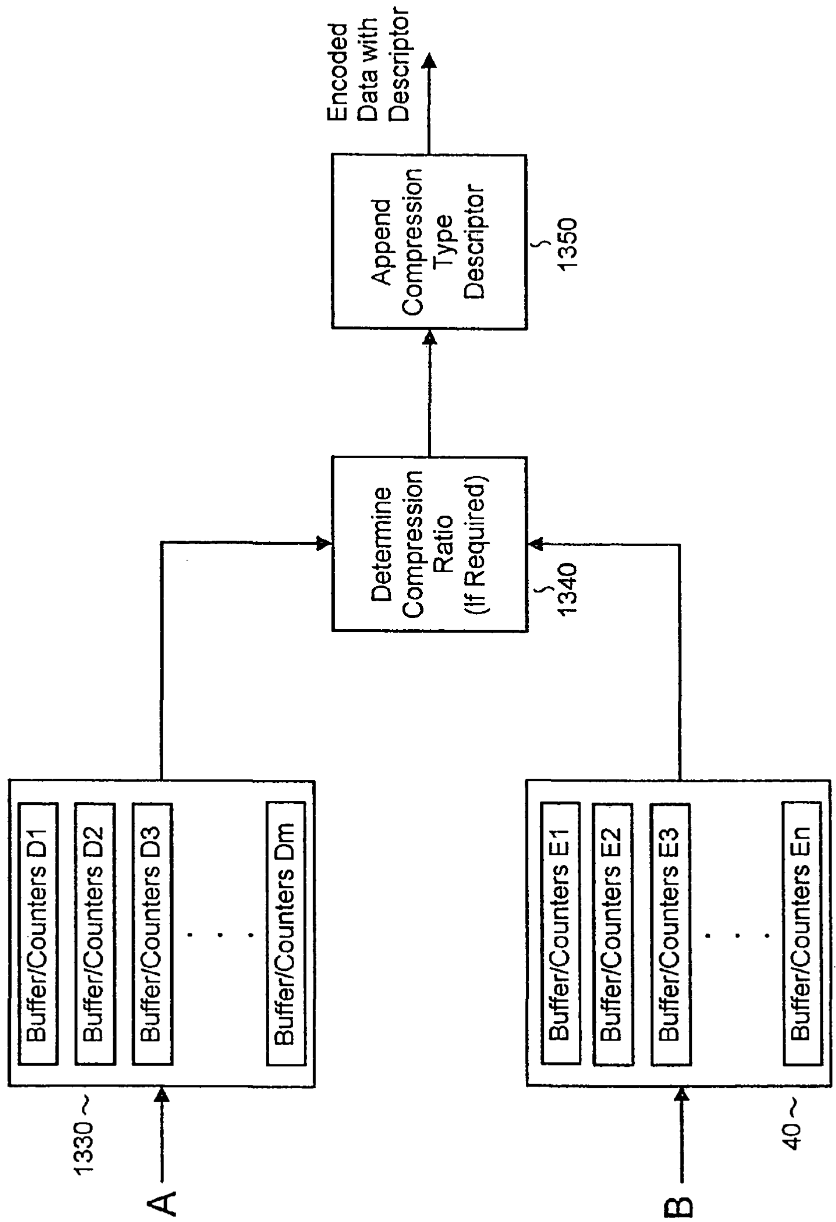
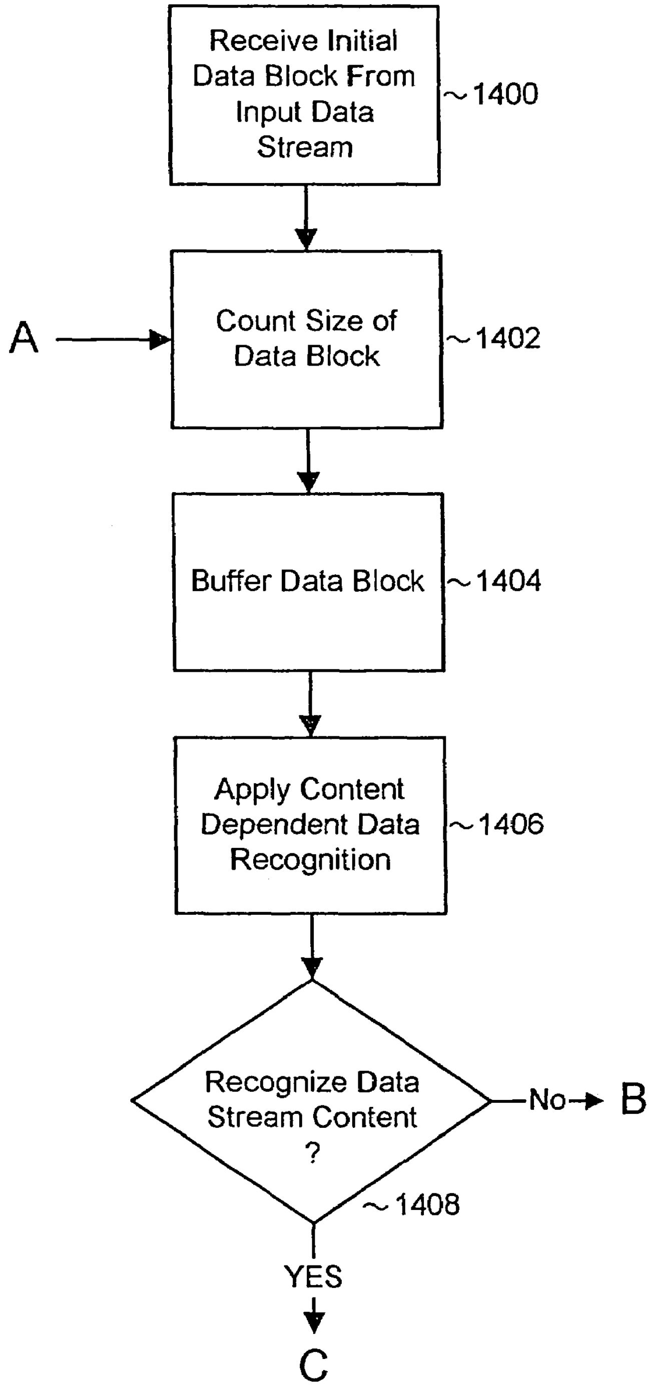

A
1330
Buffer/Counters D1
Buffer/Counters D2
Buffer/Counters D3
Buffer/Counters Dm
B
40
Buffer/Counters E1
Buffer/Counters E2
Buffer/Counters E3
Buffer/Counters En
Determine Compression Ratio (If Required)
1340
Append Compression Type Descriptor
1350
Encoded Data with Descriptor

FIGURE 13B



Receive Initial Data Block From Input Data Stream
1400
A
Count Size of Data Block
1402
Buffer Data Block
1404
Apply Content Dependent Data Recognition
1406
Recognize Data Stream Content ?
No
B
1408
YES
C

FIGURE 14A

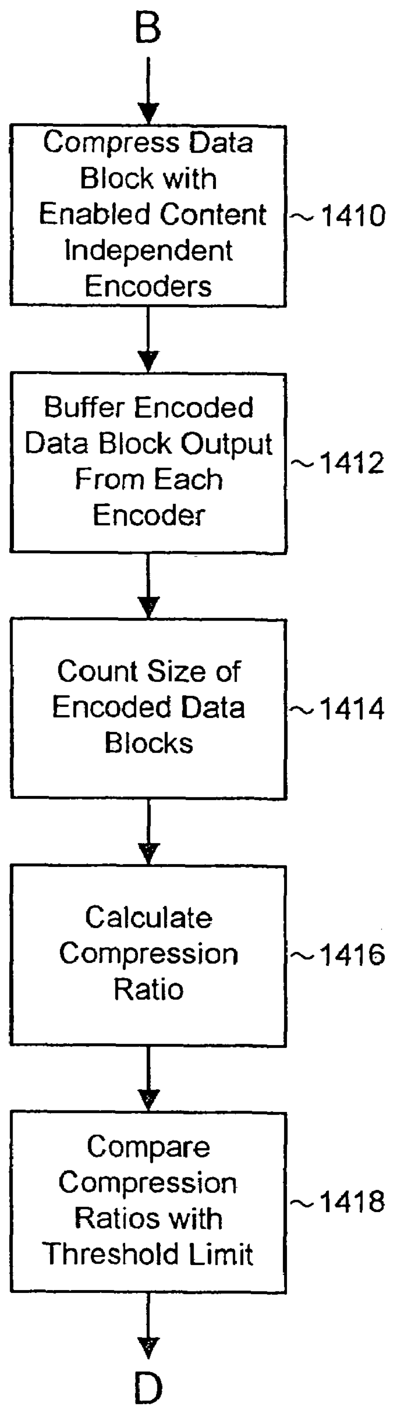

B
Compress Data Block with Enabled Content Independent Encoders
1410
Buffer Encoded Data Block Output From Each Encoder
1412
Count Size of Encoded Data Blocks
1414
Calculate Compression Ratio
1416
Compare Compression Ratios with Threshold Limit
1418
D

FIGURE 14B

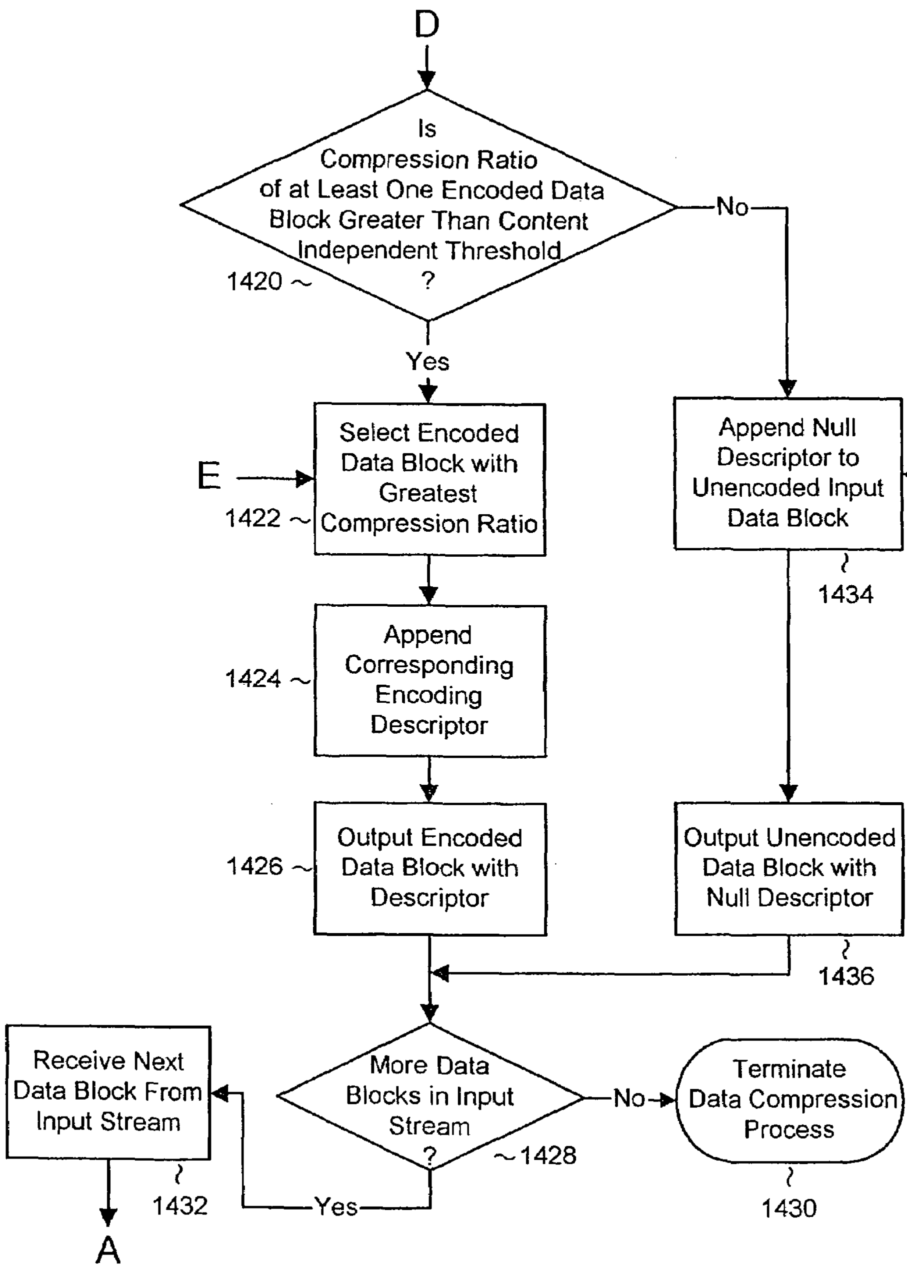

D
Is Compression Ratio of at Least One Encoded Data Block Greater Than Content Independent Threshold ?
1420
No
Yes
Select Encoded Data Block with Greatest Compression Ratio
E
1422
Append Null Descriptor to Unencoded Input Data Block
F
1434
Append Corresponding Encoding Descriptor
1424
Output Encoded Data Block with Descriptor
1426
Output Unencoded Data Block with Null Descriptor
1436
Receive Next Data Block From Input Stream
1432
More Data Blocks in Input Stream ?
1428
No
Yes
Terminate Data Compression Process
1430
A

FIGURE 14C

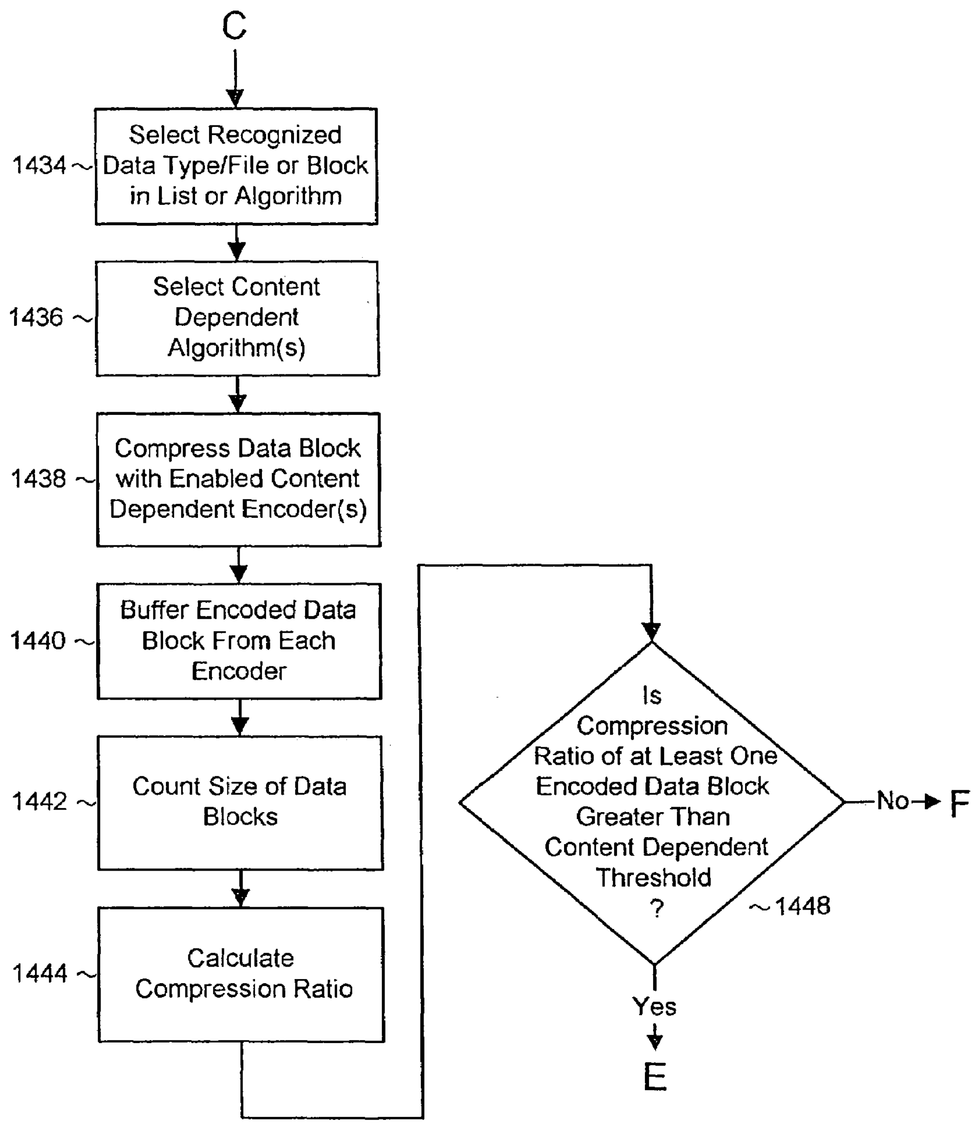

C
1434
Select Recognized Data Type/File or Block in List or Algorithm
1436
Select Content Dependent Algorithm(s)
1438
Compress Data Block with Enabled Content Dependent Encoder(s)
1440
Buffer Encoded Data Block From Each Encoder
1442
Count Size of Data Blocks
1444
Calculate Compression Ratio
Is Compression Ratio of at Least One Encoded Data Block Greater Than Content Dependent Threshold ?
1448
No
F
Yes
E

FIGURE 14D

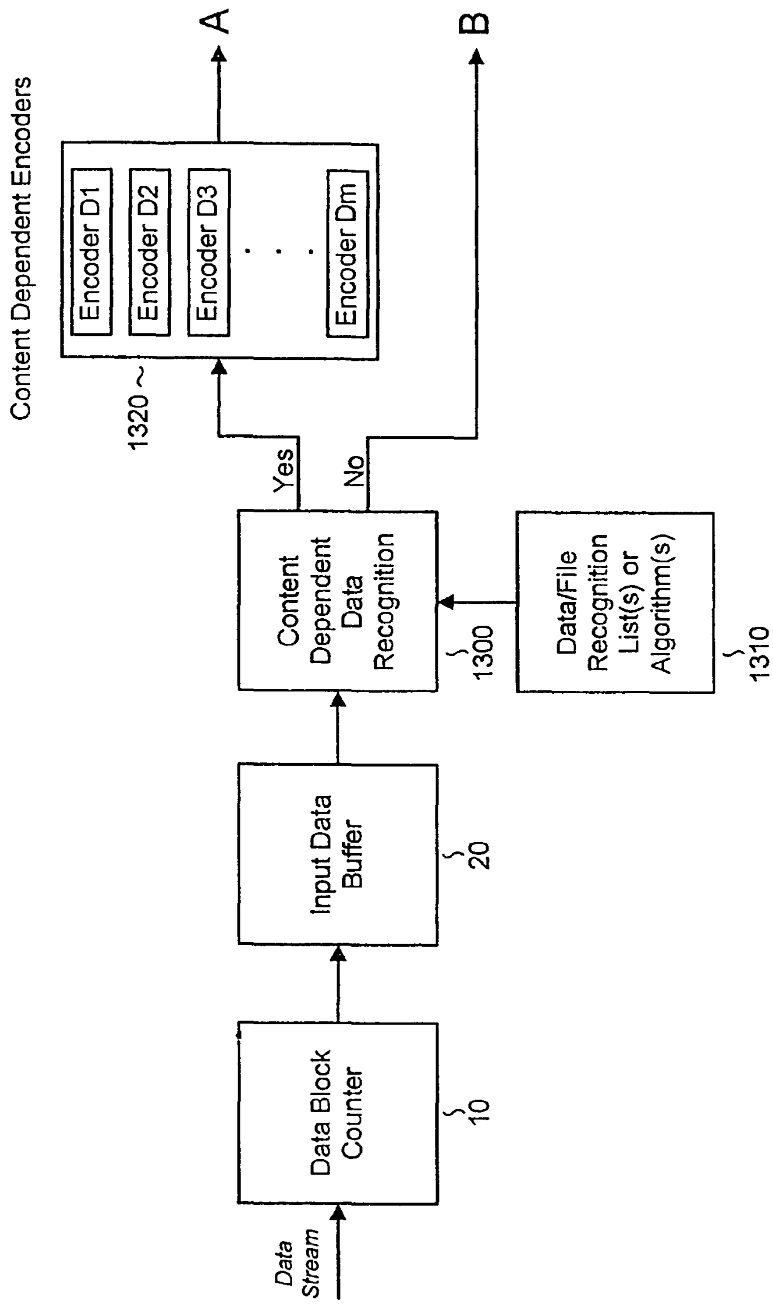

Content Dependent Encoders
1320
Encoder D1
Encoder D2
Encoder D3
Encoder Dm
A
B
Yes
No
Content Dependent Data Recognition
1300
Data/File Recognition List(s) or Algorithm(s)
1310
Input Data Buffer
20
Data Block Counter
10
Data Stream

FIGURE 15A

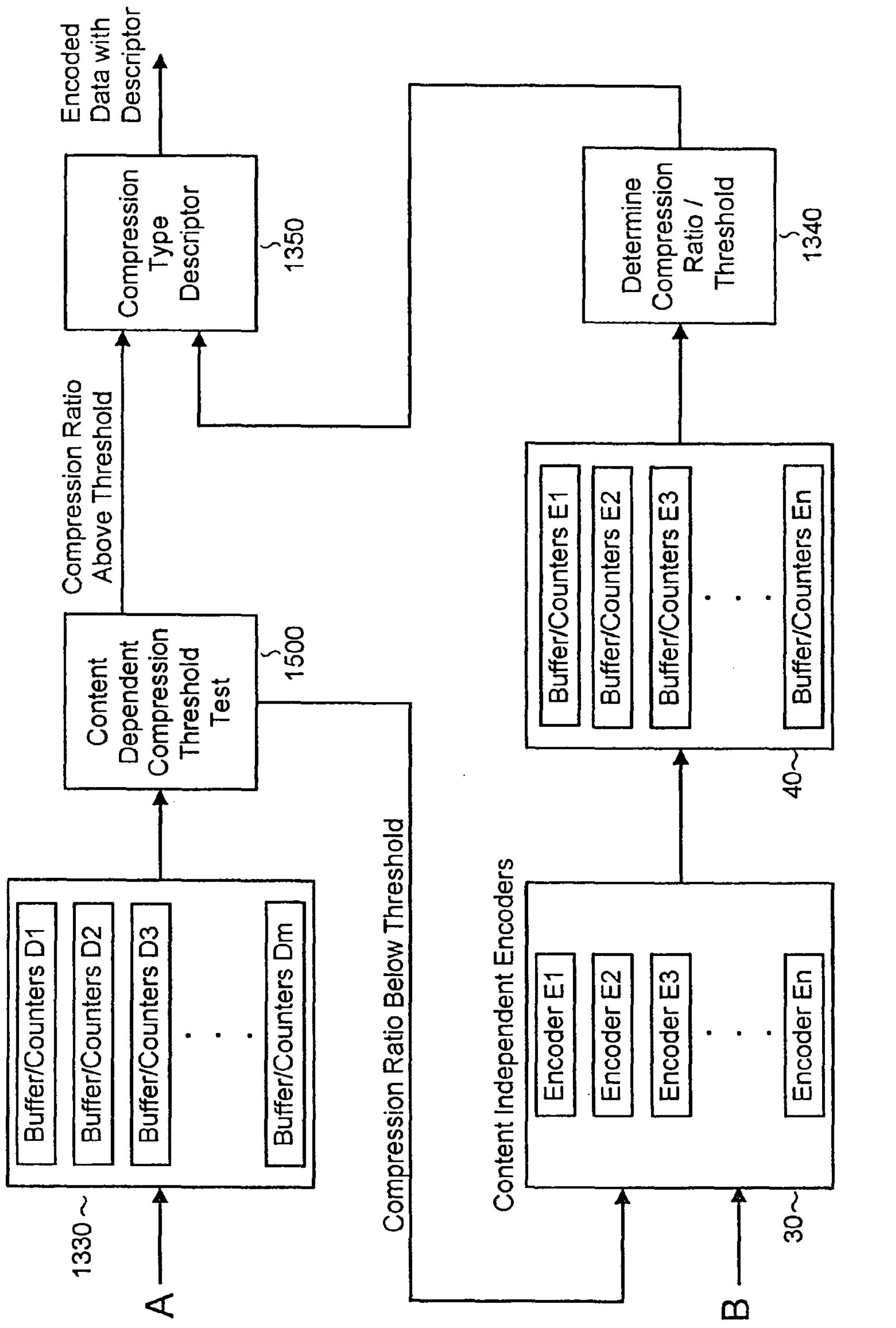

A
1330
Buffer/Counters D1
Buffer/Counters D2
Buffer/Counters D3
Buffer/Counters Dm
Content Dependent Compression Threshold Test
1500
Compression Ratio Above Threshold
Compression Ratio Below Threshold
Compression Type Descriptor
1350
Encoded Data with Descriptor
B
30
Content Independent Encoders
Encoder E1
Encoder E2
Encoder E3
Encoder En
40
Buffer/Counters E1
Buffer/Counters E2
Buffer/Counters E3
Buffer/Counters En
Determine Compression Ratio / Threshold
1340

FIGURE 15B

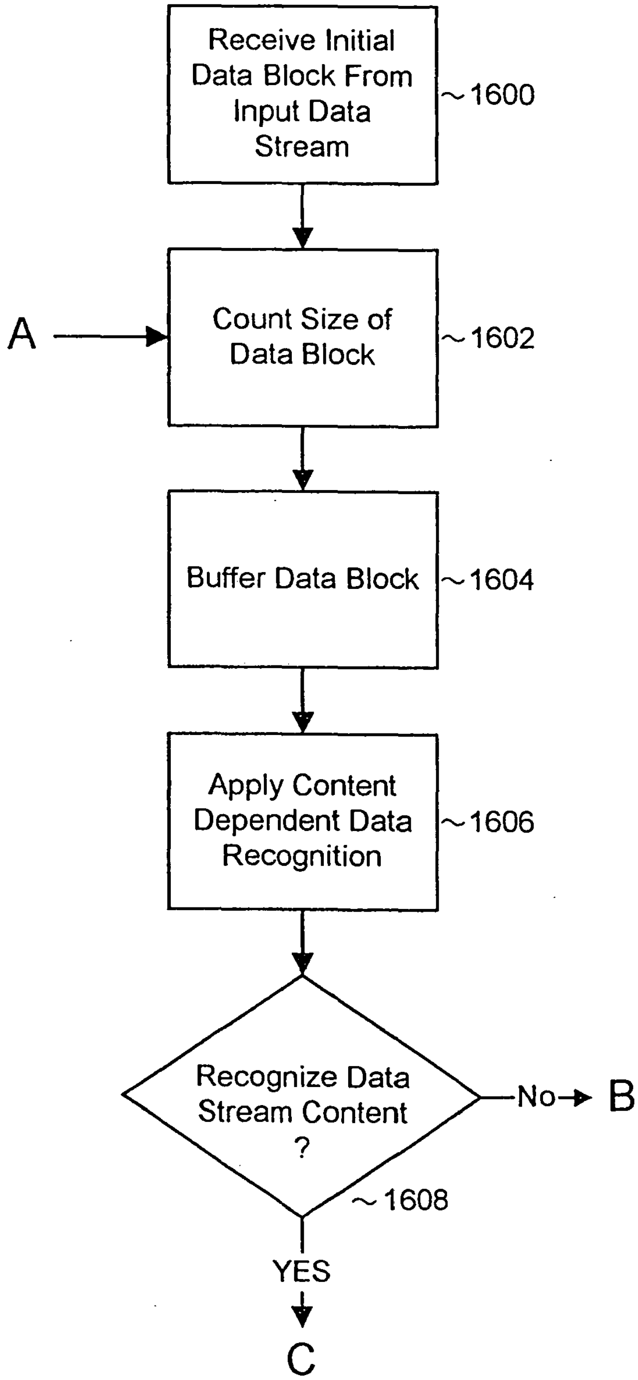

Receive Initial Data Block From Input Data Stream
1600
A
Count Size of Data Block
1602
Buffer Data Block
1604
Apply Content Dependent Data Recognition
1606
Recognize Data Stream Content ?
No
B
1608
YES
C

FIGURE 16A

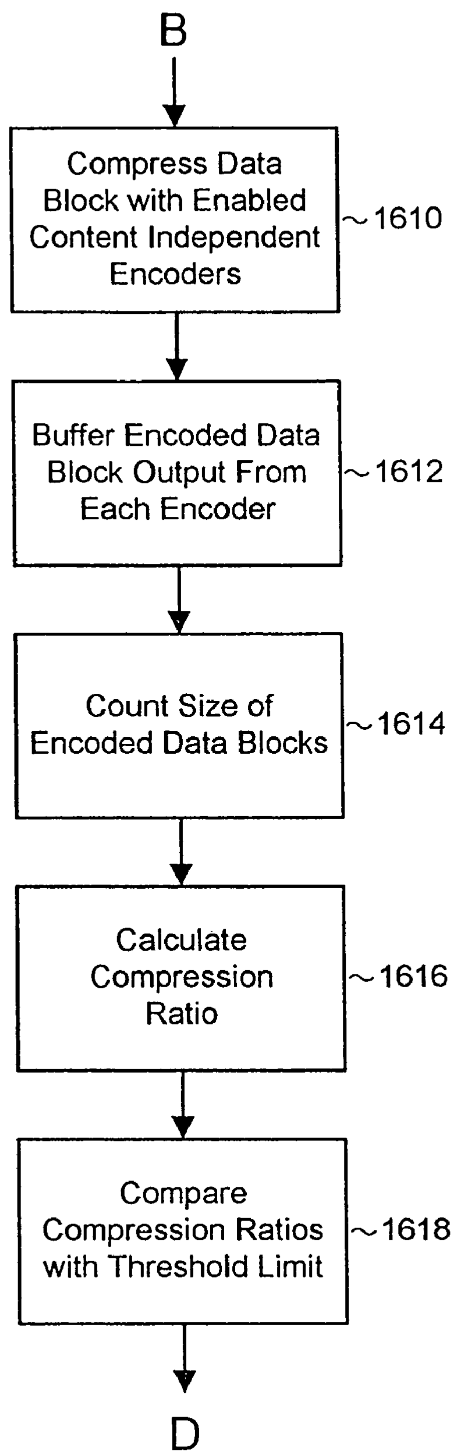

B
Compress Data Block with Enabled Content Independent Encoders
1610
Buffer Encoded Data Block Output From Each Encoder
1612
Count Size of Encoded Data Blocks
1614
Calculate Compression Ratio
1616
Compare Compression Ratios with Threshold Limit
1618
D

FIGURE 16B

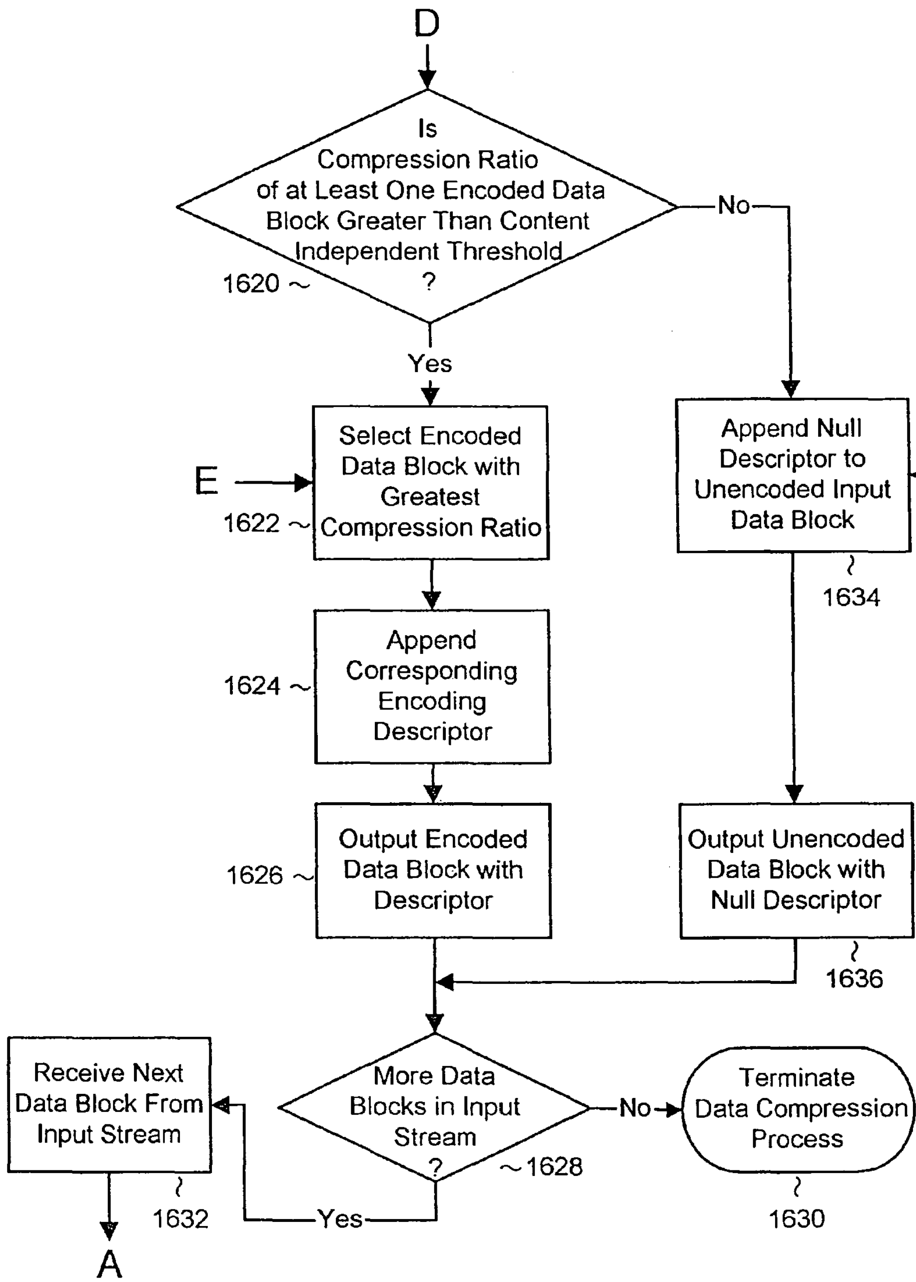

D
Is Compression Ratio of at Least One Encoded Data Block Greater Than Content Independent Threshold ?
1620
No
Yes
Select Encoded Data Block with Greatest Compression Ratio
E
1622
Append Null Descriptor to Unencoded Input Data Block
F
1634
Append Corresponding Encoding Descriptor
1624
Output Encoded Data Block with Descriptor
1626
Output Unencoded Data Block with Null Descriptor
1636
Receive Next Data Block From Input Stream
1632
More Data Blocks in Input Stream ?
1628
No
Yes
Terminate Data Compression Process
1630
A

FIGURE 16C

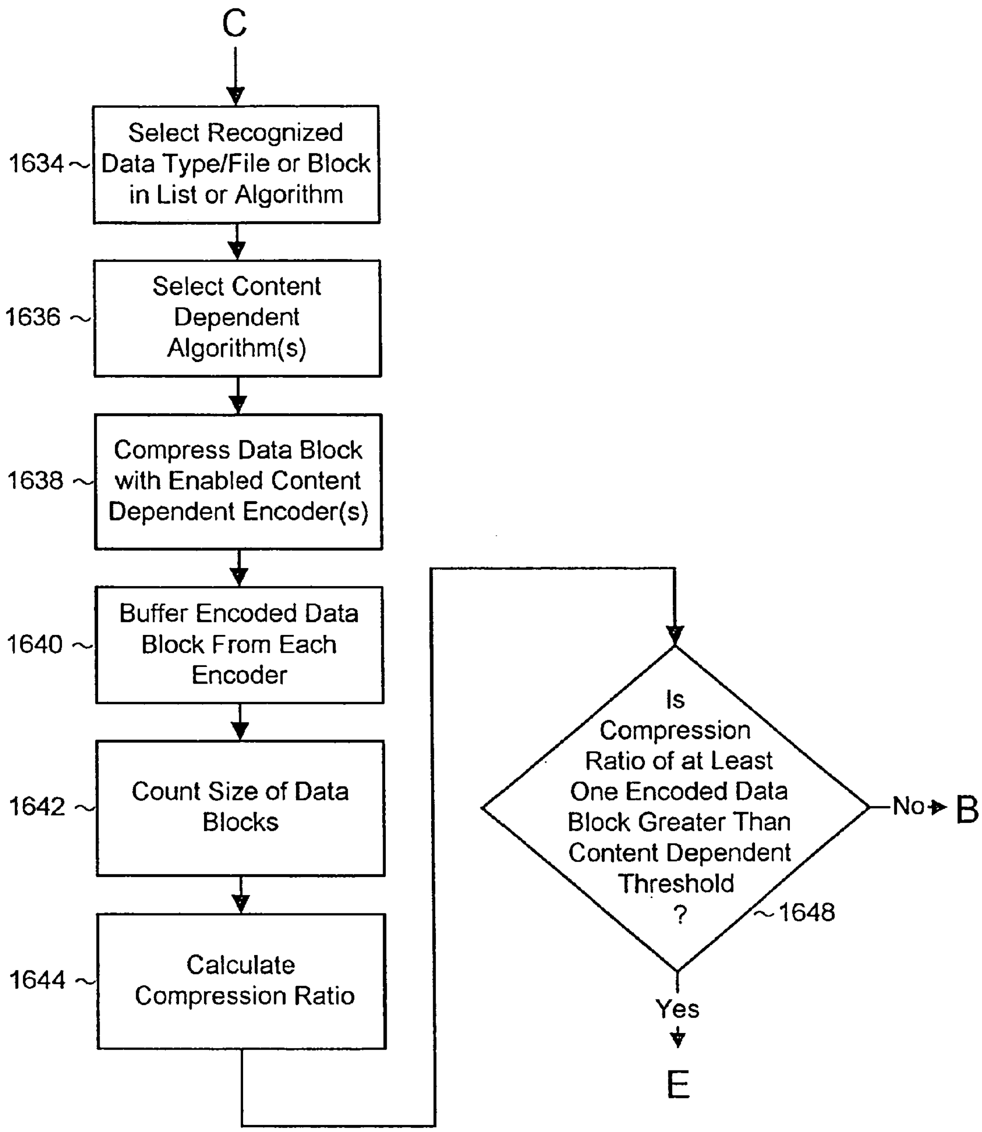

C
1634
Select Recognized Data Type/File or Block in List or Algorithm
1636
Select Content Dependent Algorithm(s)
1638
Compress Data Block with Enabled Content Dependent Encoder(s)
1640
Buffer Encoded Data Block From Each Encoder
1642
Count Size of Data Blocks
1644
Calculate Compression Ratio
Is Compression Ratio of at Least One Encoded Data Block Greater Than Content Dependent Threshold ?
1648
No
B
Yes
E

FIGURE 16D

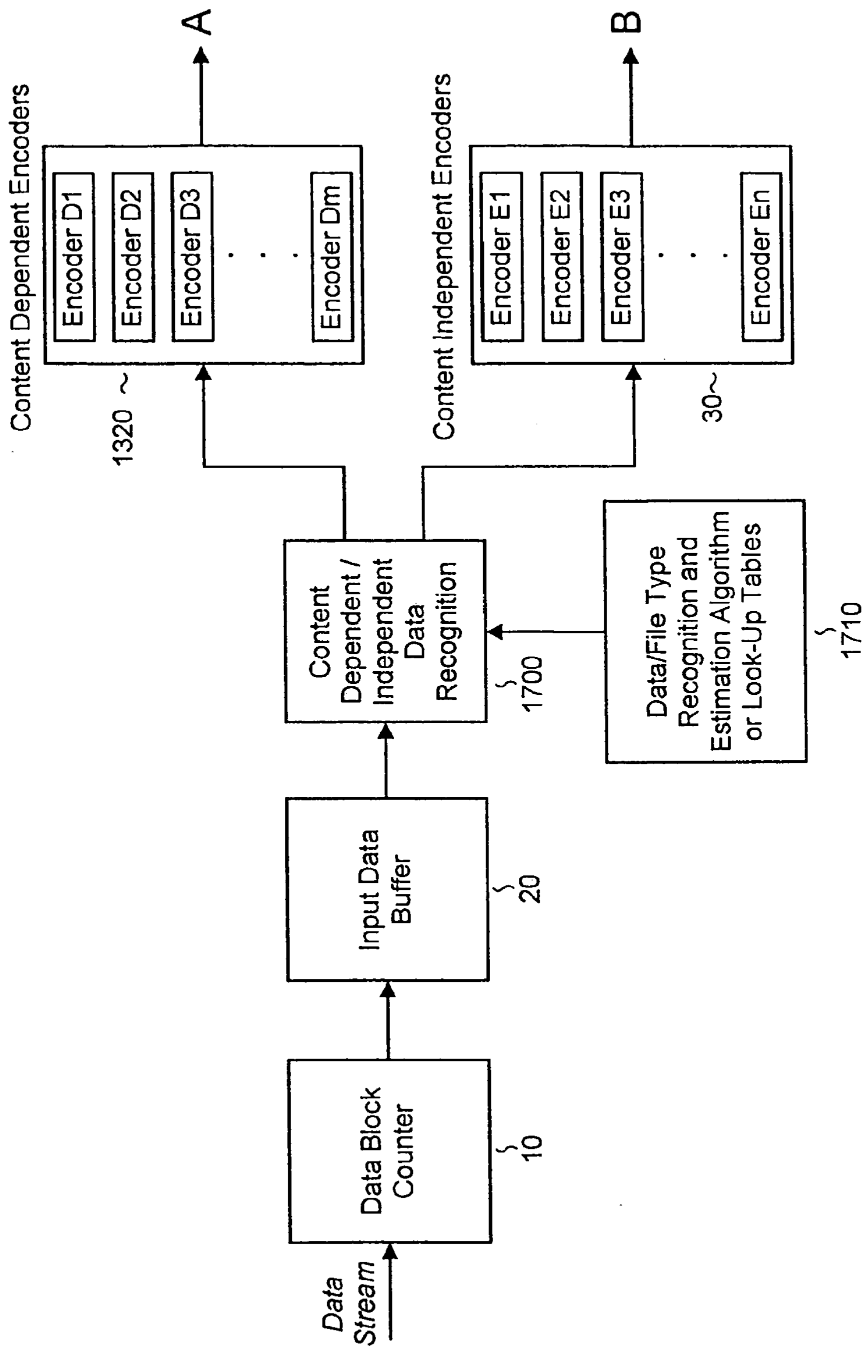

A
B
Content Dependent Encoders
Encoder D1
Encoder D2
Encoder D3
Encoder Dm
1320
Content Independent Encoders
Encoder E1
Encoder E2
Encoder E3
Encoder En
30
Content Dependent / Independent Data Recognition
1700
Data/File Type Recognition and Estimation Algorithm or Look-Up Tables
1710
Input Data Buffer
20
Data Block Counter
10
Data Stream

FIGURE 17A

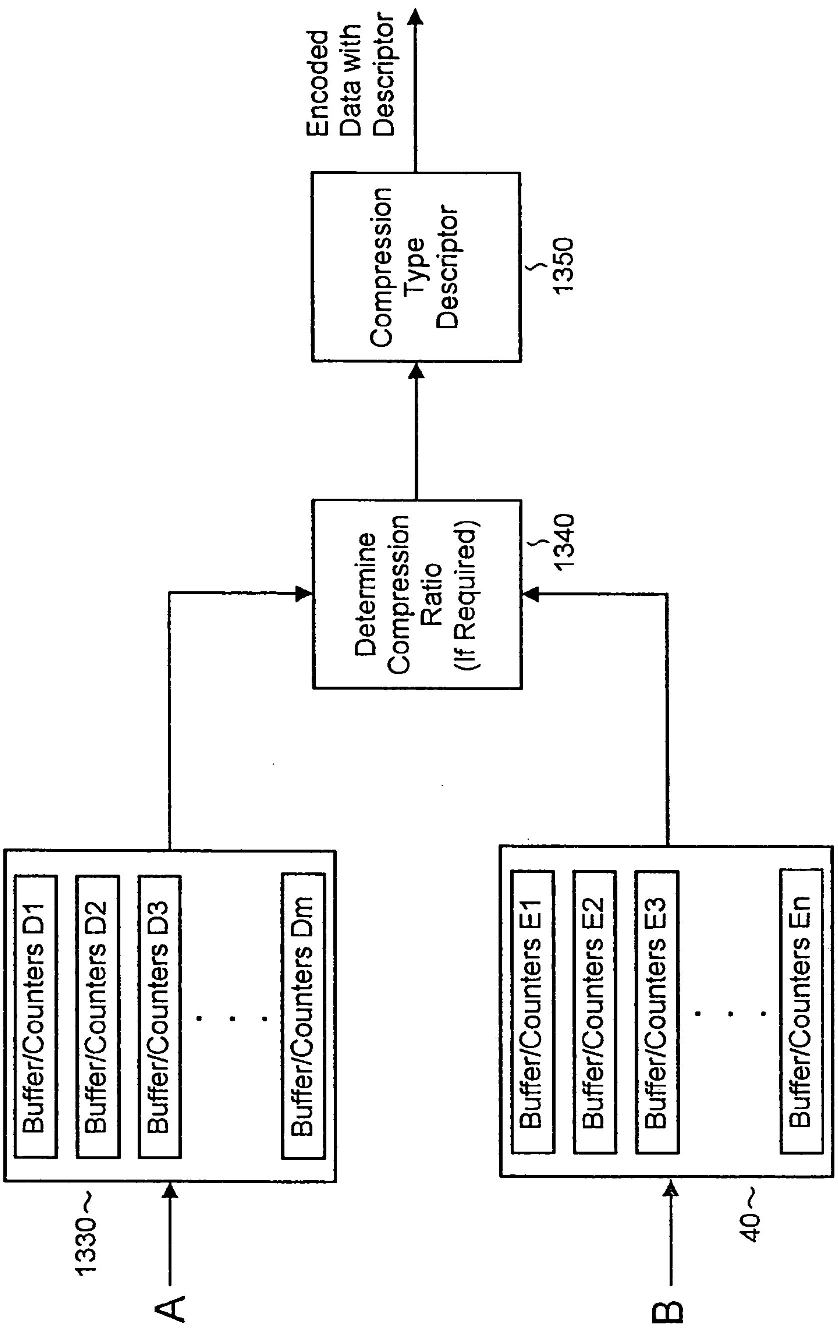

Encoded Data with Descriptor
Compression Type Descriptor
1350
Determine Compression Ratio (If Required)
1340
A
1330
Buffer/Counters D1
Buffer/Counters D2
Buffer/Counters D3
Buffer/Counters Dm
B
40
Buffer/Counters E1
Buffer/Counters E2
Buffer/Counters E3
Buffer/Counters En

FIGURE 17B

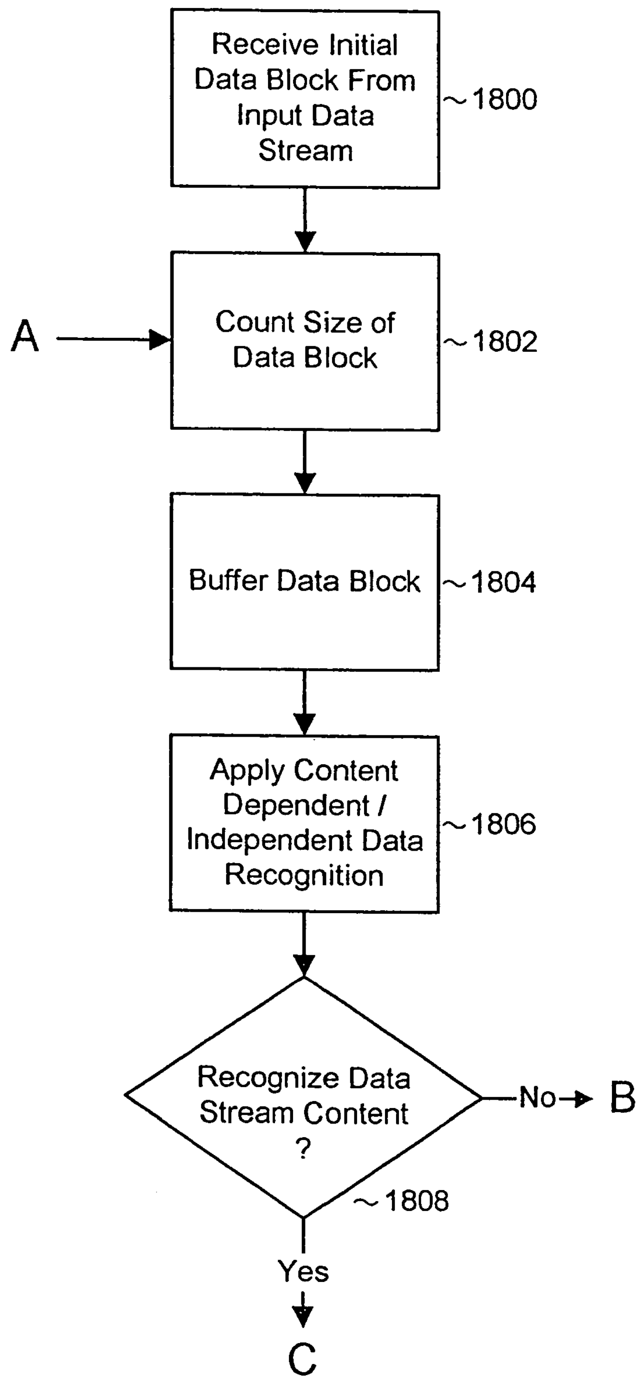
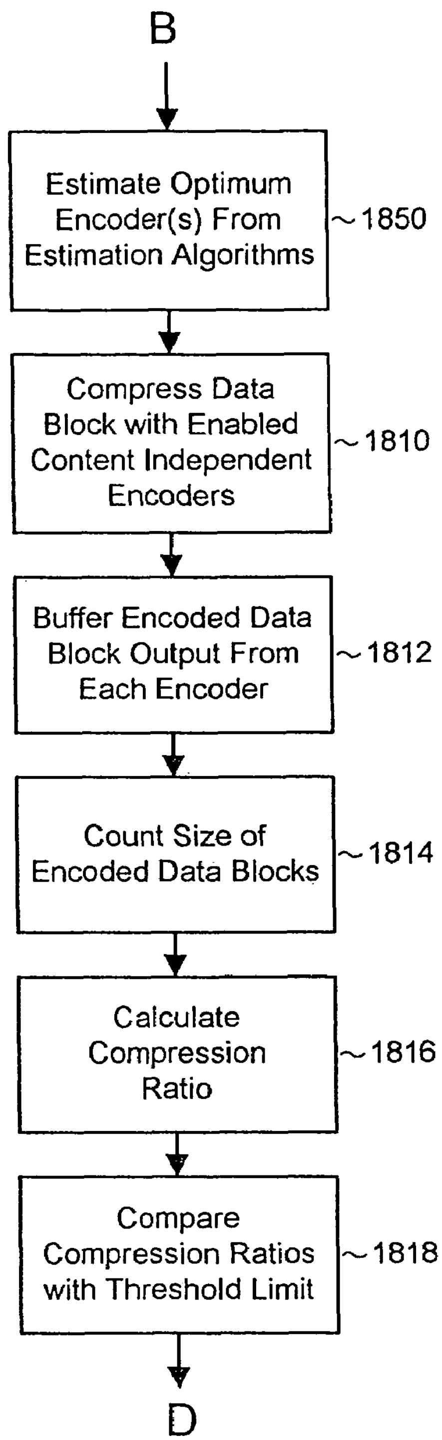
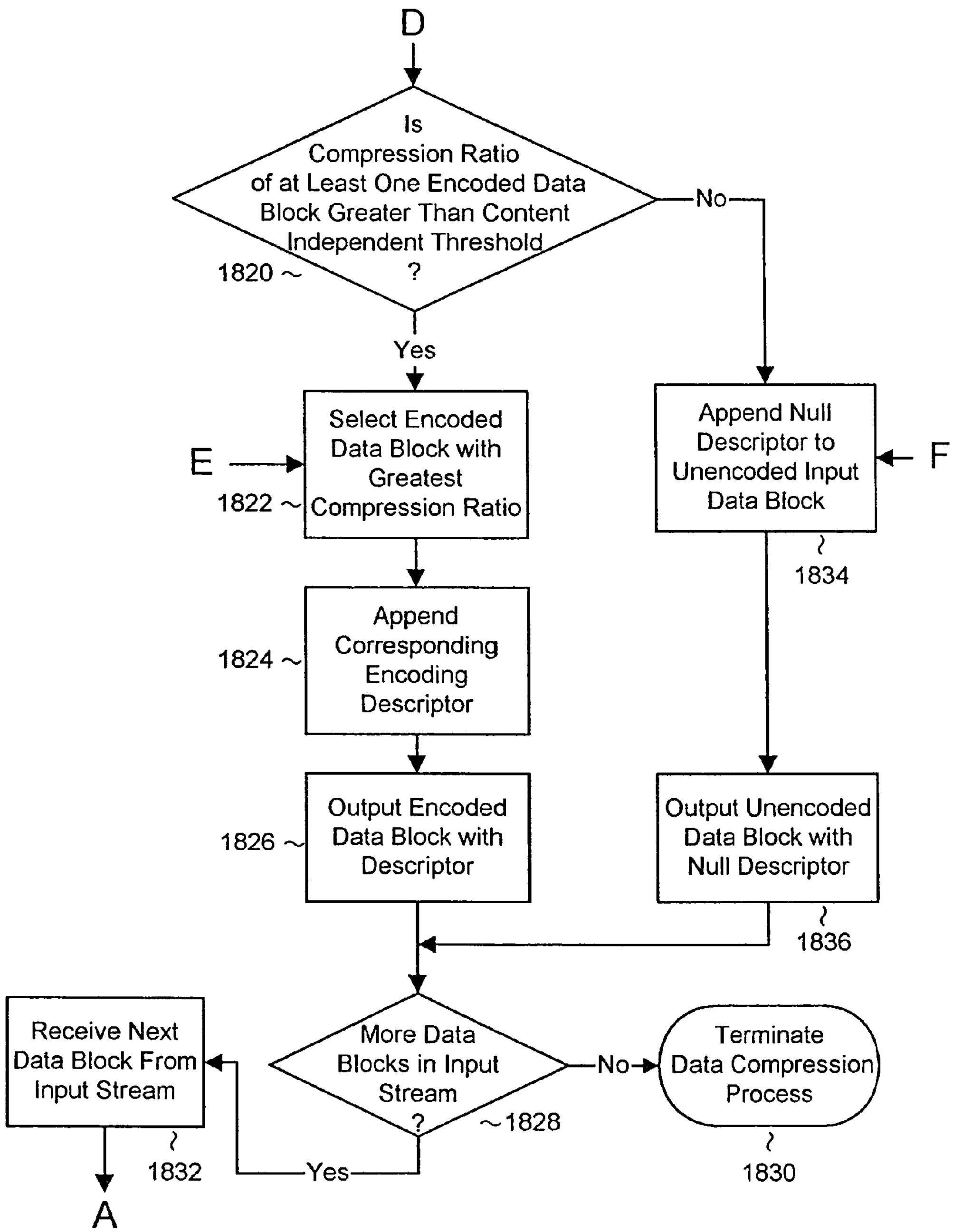

Receive Initial Data Block From Input Data Stream
1800
A
Count Size of Data Block
1802
Buffer Data Block
1804
Apply Content Dependent / Independent Data Recognition
1806
Recognize Data Stream Content ?
No
B
1808
Yes
C

FIGURE 18A



B
Estimate Optimum Encoder(s) From Estimation Algorithms
1850
Compress Data Block with Enabled Content Independent Encoders
1810
Buffer Encoded Data Block Output From Each Encoder
1812
Count Size of Encoded Data Blocks
1814
Calculate Compression Ratio
1816
Compare Compression Ratios with Threshold Limit
1818
D

FIGURE 18B



D
Is Compression Ratio of at Least One Encoded Data Block Greater Than Content Independent Threshold ?
1820
No
Yes
E
Select Encoded Data Block with Greatest Compression Ratio
1822
Append Null Descriptor to Unencoded Input Data Block
F
1834
Append Corresponding Encoding Descriptor
1824
Output Encoded Data Block with Descriptor
1826
Output Unencoded Data Block with Null Descriptor
1836
Receive Next Data Block From Input Stream
1832
More Data Blocks in Input Stream ?
1828
No
Yes
Terminate Data Compression Process
1830
A

FIGURE 18C

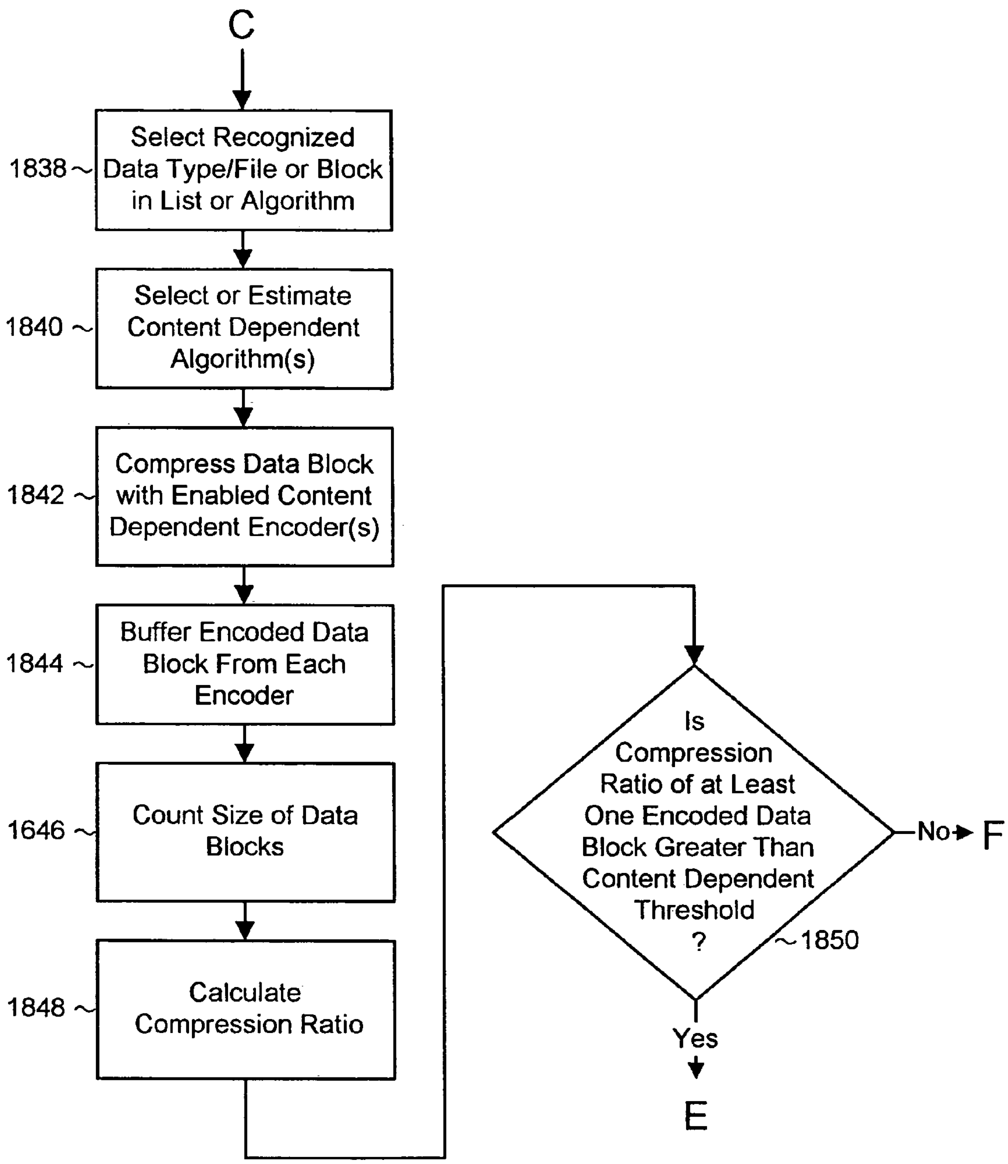

C
1838
Select Recognized Data Type/File or Block in List or Algorithm
1840
Select or Estimate Content Dependent Algorithm(s)
1842
Compress Data Block with Enabled Content Dependent Encoder(s)
1844
Buffer Encoded Data Block From Each Encoder
1646
Count Size of Data Blocks
1848
Calculate Compression Ratio
Is Compression Ratio of at Least One Encoded Data Block Greater Than Content Dependent Threshold ?
1850
No
F
Yes
E

FIGURE 18D

DATA COMPRESSION SYSTEMS AND METHODS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Continuation of U.S. patent application Ser. No. 10/668,768, which is a Continuation of U.S. patent application Ser. No. 10/016,355, now U.S. Pat. No. 6,624,761, which is a Continuation-In-Part of U.S. patent application Ser. No. 09/705,446, now U.S. Pat. No. 6,309,424, which is a Continuation of U.S. patent application Ser. No. 09/210,491, now U.S. Pat. No. 6,195,024, each of which is hereby incorporated by reference herein in its entirety.

BACKGROUND

1. Technical Field

The present invention relates generally to a data compression and decompression and, more particularly, to systems and methods for data compression using content independent and content dependent data compression and decompression.

2. Description of Related Art

Information may be represented in a variety of manners. Discrete information such as text and numbers are easily represented in digital data. This type of data representation is known as symbolic digital data. Symbolic digital data is thus an absolute representation of data such as a letter, figure, character, mark, machine code, or drawing,

Continuous information such as speech, music, audio, images and video, frequently exists in the natural world as analog information. As is well known to those skilled in the art, recent advances in very large scale integration (VLSI) digital computer technology have enabled both discrete and analog information to be represented with digital data. Continuous information represented as digital data is often referred to as diffuse data. Diffuse digital data is thus a representation of data that is of low information density and is typically not easily recognizable to humans in its native form.

There are many advantages associated with digital data representation. For instance, digital data is more readily processed, stored, and transmitted due to its inherently high noise immunity. In addition, the inclusion of redundancy in digital data representation enables error detection and/or correction. Error detection and/or correction capabilities are dependent upon the amount and type of data redundancy, available error detection and correction processing, and extent of data corruption.

One outcome of digital data representation is the continuing need for increased capacity in data processing, storage, and transmittal. This is especially true for diffuse data where increases in fidelity and resolution create exponentially greater quantities of data. Data compression is widely used to reduce the amount of data required to process, transmit, or store a given quantity of information. In general, there are two types of data compression techniques that may be utilized either separately or jointly to encode/decode data: lossless and lossy data compression.

Lossy data compression techniques provide for an inexact representation of the original uncompressed data such that the decoded (or reconstructed) data differs from the original unencoded/uncompressed data. Lossy data compression is also known as irreversible or noisy compression. Entropy is defined as the quantity of information in a given set of data. Thus, one obvious advantage of lossy data compression is that the compression ratios can be larger than the entropy limit, all at the expense of information content. Many lossy data compression techniques seek to exploit various traits within the human senses to eliminate otherwise imperceptible data. For example, lossy data compression of visual imagery might seek to delete information content in excess of the display resolution or contrast ratio.

On the other hand, lossless data compression techniques provide an exact representation of the original uncompressed data. Simply stated, the decoded (or reconstructed) data is identical to the original unencoded/uncompressed data. Lossless data compression is also known as reversible or noiseless compression. Thus, lossless data compression has, as its current limit, a minimum representation defined by the entropy of a given data set.

There are various problems associated with the use of lossless compression techniques. One fundamental problem encountered with most lossless data compression techniques are their content sensitive behavior. This is often referred to as data dependency. Data dependency implies that the compression ratio achieved is highly contingent upon the content of the data being compressed. For example, database files often have large unused fields and high data redundancies, offering the opportunity to losslessly compress data at ratios of 5 to 1 or more. In contrast, concise software programs have little to no data redundancy and, typically, will not losslessly compress better than 2 to 1.

Another problem with lossless compression is that there are significant variations in the compression ratio obtained when using a single lossless data compression technique for data streams having different data content and data size. This process is known as natural variation.

A further problem is that negative compression may occur when certain data compression techniques act upon many types of highly compressed data. Highly compressed data appears random and many data compression techniques will substantially expand, not compress this type of data.

For a given application, there are many factors that govern the applicability of various data compression techniques. These factors include compression ratio, encoding and decoding processing requirements, encoding and decoding time delays, compatibility with existing standards, and implementation complexity and cost, along with the adaptability and robustness to variations in-input data. A direct relationship exists in the current art between compression ratio and the amount and complexity of processing required. One of the limiting factors in most existing prior art lossless data compression techniques is the rate at which the encoding and decoding processes are performed. Hardware and software implementation tradeoffs are often dictated by encoder and decoder complexity along with cost.

Another problem associated with lossless compression methods is determining the optimal compression technique for a given set of input data and intended application. To combat this problem, there are many conventional content dependent techniques that may be utilized. For instance, file type descriptors are typically appended to file names to describe the application programs that normally act upon the data contained within the file. In this manner data types, data structures, and formats within a given file may be ascertained. Fundamental limitations with this content dependent technique include:

(1) the extremely large number of application programs, some of which do not possess published or documented file formats, data structures, or data type descriptors;

(2) the ability for any data compression supplier or consortium to acquire, store, and access the vast amounts of data required to identify known file descriptors and associated data types, data structures, and formats; and

(3) the rate at which new application programs are developed and the need to update file format data descriptions accordingly.

An alternative technique that approaches the problem of selecting an appropriate lossless data compression technique is disclosed, for example, in U.S. Pat. No. 5,467,087 to Chu entitled "High Speed Lossless Data Compression System" ("Chu"). FIG. **1** illustrates an embodiment of this data compression and decompression technique. Data compression **1** comprises two phases, a data pre-compression phase **2** and a data compression phase **3**. Data decompression **4** of a compressed input data stream is also comprised of two phases, a data type retrieval phase **5** and a data decompression phase **6**. During the data compression process **1**, the data pre-compressor **2** accepts an uncompressed data stream, identifies the data type of the input stream, and generates a data type identification signal. The data compressor **3** selects a data compression method from a preselected set of methods to compress the input data stream, with the intention of producing the best available compression ratio for that particular data type.

There are several limitations associated with the Chu method. One such limitation is the need to unambiguously identify various data types. While these might include such common data types as ASCII, binary, or unicode, there, in fact, exists a broad universe of data types that fall outside the three most common data types. Examples of these alternate data types include: signed and unsigned integers of various lengths, differing types and precision of floating point numbers, pointers, other forms of character text, and a multitude of user defined data types. Additionally, data types may be interspersed or partially compressed, making data type recognition difficult and/or impractical. Another limitation is that given a known data type, or mix of data types within a specific set or subset of input data, it may be difficult and/or impractical to predict which data encoding technique yields the highest compression ratio.

Accordingly, there is a need for a data compression system and method that would address limitations in conventional data compression techniques as described above.

SUMMARY OF THE INVENTION

The present invention is directed to systems and methods for providing fast and efficient data compression using a combination of content independent data compression and content dependent data compression. In one aspect of the invention, a method for compressing data comprises the steps of:

analyzing a data block of an input data stream to identify a data type of the data block, the input data stream comprising a plurality of disparate data types;

performing content dependent data compression on the data block, if the data type of the data block is identified;

performing content independent data compression on the data block, if the data type of the data block is not identified.

In another aspect, the step of performing content independent data compression comprises: encoding the data block with a plurality of encoders to provide a plurality of encoded data blocks; determining a compression ratio obtained for each of the encoders; comparing each of the determined compression ratios with a first compression threshold; selecting for output the input data block and appending a null compression descriptor to the input data block, if all of the encoder compression ratios do not meet the first compression threshold; and selecting for output the encoded data block having the highest compression ratio and appending a corresponding compression type descriptor to the selected encoded data block, if at least one of the compression ratios meet the first compression threshold.

In another aspect, the step of performing content dependent compression comprises the steps of: selecting one or more encoders associated with the identified data type and encoding the data block with the selected encoders to provide a plurality of encoded data blocks; determining a compression ratio obtained for each of the selected encoders; comparing each of the determined compression ratios with a second compression threshold; selecting for output the input data block and appending a null compression descriptor to the input data block, if all of the encoder compression do not meet the second compression threshold; and selecting for output the encoded data block having the highest compression ratio and appending a corresponding compression type descriptor to the selected encoded data block, if at least one of the compression ratios meet the second compression threshold.

In yet another aspect, the step of performing content independent data compression on the data block, if the data type of the data block is not identified, comprises the steps of: estimating a desirability of using of one or more encoder types based one characteristics of the data block; and compressing the data block using one or more desirable encoders.

In another aspect, the step of performing content dependent data compression on the data block, if the data type of the data block is identified, comprises the steps of: estimating a desirability of using of one or more encoder types based on characteristics of the data block; and compressing the data block using one or more desirable encoders.

In another aspect, the step of analyzing the data block comprises analyzing the data block to recognize one of a data type, data structure, data block format, file substructure, and/or file types. A further step comprises maintaining an association between encoder types and data types, data structures, data block formats, file substructure, and/or file types.

In yet another aspect of the invention, a method for compressing data comprises the steps of:

analyzing a data block of an input data stream to identify a data type of the data block, the input data stream comprising a plurality of disparate data types;

performing content dependent data compression on the data block, if the data type of the data block is identified;

determining a compression ratio of the compressed data block obtained using the content dependent compression and comparing the compression ratio with a first compression threshold; and

performing content independent data compression on the data block, if the data type of the data block is not identified or if the compression ratio of the compressed data block obtained using the content dependent compression does not meet the first compression threshold.

Advantageously, the present invention employs a plurality of encoders applying a plurality of compression techniques on an input data stream so as to achieve maximum compression in accordance with the real-time or pseudo real-time data rate constraint. Thus, the output bit rate is not fixed and the amount, if any, of permissible data quality degradation is user or data specified.

These and other aspects, features and advantages of the present invention will become apparent from the following

detailed description of preferred embodiments, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block/flow diagram of a content dependent high-speed lossless data compression and decompression system/method according to the prior art;

FIG. **2** is a block diagram of a content independent data compression system according to one embodiment of the present invention;

FIGS. **3***a* and **3***b* comprise a flow diagram of a data compression method according to one aspect of the present invention, which illustrates the operation of the data compression system of FIG. **2**;

FIG. **4** is a block diagram of a content independent data compression system according to another embodiment of the present invention having an enhanced metric for selecting an optimal encoding technique;

FIGS. **5***a* and **5***b* comprise a flow diagram of a data compression method according to another aspect of the present invention, which illustrates the operation of the data compression system of FIG. **4**;

FIG. **6** is a block diagram of a content independent data compression system according to another embodiment of the present invention having an a priori specified timer that provides real-time or pseudo real-time of output data;

FIGS. **7***a* and **7***b* comprise a flow diagram of a data compression method according to another aspect of the present invention, which illustrates the operation of the data compression system of FIG. **6**;

FIG. **8** is a block diagram of a content independent data compression system according to another embodiment having an a priori specified timer that provides real-time or pseudo real-time of output data and an enhanced metric for selecting an optimal encoding technique;

FIG. **9** is a block diagram of a content independent data compression system according to another embodiment of the present invention having an encoding architecture comprising a plurality of sets of serially cascaded encoders;

FIGS. **10***a* and **10***b* comprise a flow diagram of a data compression method according to another aspect of the present invention, which illustrates the operation of the data compression system of FIG. **9**;

FIG. **11** is block diagram of a content independent data decompression system according to one embodiment of the present invention;

FIG. **12** is a flow diagram of a data decompression method according to one aspect of the present invention, which illustrates the operation of the data compression system of FIG. **11**;

FIGS. **13***a* and **13***b* comprise a block diagram of a data compression system comprising content dependent and content independent data compression, according to an embodiment of the present invention;

FIGS. **14***a*-**14***d* comprise a flow diagram of a data compression method using both content dependent and content independent data compression, according to one aspect of the present invention;

FIGS. **15***a* and **15***b* comprise a block diagram of a data compression system comprising content dependent and content independent data compression, according to another embodiment of the present invention;

FIGS. **16***a*-**16***d* comprise a flow diagram of a data compression method using both content dependent and content independent data compression, according to another aspect of the present invention;

FIGS. **17***a* and **17***b* comprise a block diagram of a data compression system comprising content dependent and content independent data compression, according to another embodiment of the present invention; and

FIGS. **18***a*-**18***d* comprise a flow diagram of a data compression method using both content dependent and content independent data compression, according to another aspect of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention is directed to systems and methods for providing data compression and decompression using content independent and content dependent data compression and decompression. In the following description, it is to be understood that system elements having equivalent or similar functionality are designated with the same reference numerals in the Figures. It is to be further understood that the present invention may be implemented in various forms of hardware, software, firmware, or a combination thereof. In particular, the system modules described herein are preferably implemented in software as an application program that is executable by, e.g., a general purpose computer or any machine or device having any suitable and preferred microprocessor architecture. Preferably, the present invention is implemented on a computer platform including hardware such as one or more central processing units (CPU), a random access memory (RAM), and input/output (I/O) interface(s). The computer platform also includes an operating system and microinstruction code. The various processes and functions described herein may be either part of the microinstruction code or application programs which are executed via the operating system. In addition, various other peripheral devices may be connected to the computer platform such as an additional data storage device and a printing device.

It is to be further understood that, because some of the constituent system components described herein are preferably implemented as software modules, the actual system connections shown in the Figures may differ depending upon the manner in which the systems are programmed. It is to be appreciated that special purpose microprocessors may be employed to implement the present invention. Given the teachings herein, one of ordinary skill in the related art will be able to contemplate these and similar implementations or configurations of the present invention.

Referring now to FIG. **2** a block diagram illustrates a content independent data compression system according to one embodiment of the present invention. The data compression system includes a counter module **10** that receives as input an uncompressed or compressed data stream. It is to be understood that the system processes the input data stream in data blocks that may range in size from individual bits through complete files or collections of multiple files. Additionally, the data block size may be fixed or variable. The counter module **10** counts the size of each input data block (i.e., the data block size is counted in bits, bytes, words, any convenient data multiple or metric, or any combination thereof).

An input data buffer **20**, operatively connected to the counter module **10**, may be provided for buffering the input data stream in order to output an uncompressed data stream in the event that, as discussed in further detail below, every encoder fails to achieve a level of compression that exceeds an a priori specified minimum compression ratio threshold. It is to be understood that the input data buffer **20** is not required for implementing the present invention.

An encoder module **30** is operatively connected to the buffer **20** and comprises a set of encoders E**1**, E**2**, E**3** . . . En. The encoder set E**1**, E**2**, E**3** . . . En may include any number "n" of those lossless encoding techniques currently well known within the art such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. It is to be understood that the encoding techniques are selected based upon their ability to effectively encode different types of input data. It is to be appreciated that a full complement of encoders are preferably selected to provide a broad coverage of existing and future data types.

The encoder module **30** successively receives as input each of the buffered input data blocks (or unbuffered input data blocks from the counter module **10**). Data compression is performed by the encoder module **30** wherein each of the encoders E**1** . . . En processes a given input data block and outputs a corresponding set of encoded data blocks. It is to be appreciated that the system affords a user the option to enable/disable any one or more of the encoders E**1** . . . En prior to operation. As is understood by those skilled in the art, such feature allows the user to tailor the operation of the data compression system for specific applications. It is to be further appreciated that the encoding process may be performed either in parallel or sequentially. In particular, the encoders E**1** through En of encoder module **30** may operate in parallel (i.e., simultaneously processing a given input data block by utilizing task multiplexing on a single central processor, via dedicated hardware, by executing on a plurality of processor or dedicated hardware systems, or any combination thereof). In addition, encoders E**1** through En may operate sequentially on a given unbuffered or buffered input data block. This process is intended to eliminate the complexity and additional processing overhead associated with multiplexing concurrent encoding techniques on a single central processor and/or dedicated hardware, set of central processors and/or dedicated hardware, or any achievable combination. It is to be further appreciated that encoders of the identical type may be applied in parallel to enhance encoding speed. For instance, encoder E**1** may comprise two parallel Huffman encoders for parallel processing of an input data block.

A buffer/counter module **40** is operatively connected to the encoding module **30** for buffering and counting the size of each of the encoded data blocks output from encoder module **30**. Specifically, the buffer/counter **30** comprises a plurality of buffer/counters BC**1**, BC**2**, BC**3** . . . BCn, each operatively associated with a corresponding one of the encoders E**1** . . . En. A compression ratio module **50**, operatively connected to the output buffer/counter **40**, determines the compression ratio obtained for each of the enabled encoders E**1** . . . En by taking the ratio of the size of the input data block to the size of the output data block stored in the corresponding buffer/counters BC**1** . . . BCn. In addition, the compression ratio module **50** compares each compression ratio with an a priori-specified compression ratio threshold limit to determine if at least one of the encoded data blocks output from the enabled encoders E**1** . . . En achieves a compression that exceeds an a priori-specified threshold. As is understood by those skilled in the art, the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. A description module **60**, operatively coupled to the compression ratio module **50**, appends a corresponding compression type descriptor to each encoded data block which is selected for output so as to indicate the type of compression format of the encoded data block.

The operation of the data compression system of FIG. **2** will now be discussed in further detail with reference to the flow diagram of FIGS. **3***a* and **3***b*. A data stream comprising one or more data blocks is input into the data compression system and the first data block in the stream is received (step **300**). As stated above, data compression is performed on a per data block basis. Accordingly, the first input data block in the input data stream is input into the counter module **10** that counts the size of the data block (step **302**). The data block is then stored in the buffer **20** (step **304**). The data block is then sent to the encoder module **30** and compressed by each (enabled) encoder E**1** . . . En (step **306**). Upon completion of the encoding of the input data block, an encoded data block is output from each (enabled) encoder E**1** . . . En and maintained in a corresponding buffer (step **308**), and the encoded data block size is counted (step **310**).

Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10**) to the size of each encoded data block output from the enabled encoders (step **312**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **314**). It is to be understood that the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It is to be further understood that notwithstanding that the current limit for lossless data compression is the entropy limit (the present definition of information content) for the data, the present invention does not preclude the use of future developments in lossless data compression that may increase lossless data compression ratios beyond what is currently known within the art.

After the compression ratios are compared with the threshold, a determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **316**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **316**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **318**). A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the in put data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **320**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **316**), then the encoded data block having the greatest compression ratio is selected (step **322**). An appropriate data compression type descriptor is then appended (step **324**). A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **326**).

After the encoded data block or the unencoded data input data block is output (steps **326** and **320**), a determination is made as to whether the input data stream contains additional

data blocks to be processed (step **328**). If the input data stream includes additional data blocks (affirmative result in step **328**), the next successive data block is received (step **330**), its block size is counted (return to step **302**) and the data compression process in repeated. This process is iterated for each data block in the input data stream. Once the final input data block is processed (negative result in step **328**), data compression of the input data stream is finished (step **322**).

Since a multitude of data types may be present within a given input data block, it is often difficult and/or impractical to predict the level of compression that will be achieved by a specific encoder. Consequently, by processing the input data blocks with a plurality of encoding techniques and comparing the compression results, content free data compression is advantageously achieved. It is to be appreciated that this approach is scalable through future generations of processors, dedicated hardware, and software. As processing capacity increases and costs reduce, the benefits provided by the present invention will continue to increase. It should again be noted that the present invention may employ any lossless data encoding technique.

Referring now to FIG. **4**, a block diagram illustrates a content independent data compression system according to another embodiment of the present invention. The data compression system depicted in FIG. **4** is similar to the data compression system of FIG. **2** except that the embodiment of FIG. **4** includes an enhanced metric functionality for selecting an optimal encoding technique. In particular, each of the encoders E**1** . . . En in the encoder module **30** is tagged with a corresponding one of user-selected encoder desirability factors **70**. Encoder desirability is defined as an a priori user specified factor that takes into account any number of user considerations including, but not limited to, compatibility of the encoded data with existing standards, data error robustness, or any other aggregation of factors that the user wishes to consider for a particular application. Each encoded data block output from the encoder module **30** has a corresponding desirability factor appended thereto. A figure of merit module **80**, operatively coupled to the compression ratio module **50** and the descriptor module **60**, is provided for calculating a figure of merit for each of the encoded data blocks which possess a compression ratio greater than the compression ratio threshold limit. The figure of merit for each encoded data block is comprised of a weighted average of the a priori user specified threshold and the corresponding encoder desirability factor. As discussed below in further detail with reference to FIGS. **5***a* and **5***b*, the figure of merit substitutes the a priori user compression threshold limit for selecting and outputting encoded data blocks.

The operation of the data compression system of FIG. **4** will now be discussed in further detail with reference to the flow diagram of FIGS. **5***a* and **5***b*. A data stream comprising one or more data blocks is input into the data compression system and the first data block in the stream is received (step **500**). The size of the first data block is then determined by the counter module **10** (step **502**). The data block is then stored in the buffer **20** (step **504**). The data block is then sent to the encoder module **30** and compressed by each (enabled) encoder in the encoder set E**1** . . . En (step **506**). Each encoded data block processed in the encoder module **30** is tagged with an encoder desirability factor that corresponds the particular encoding technique applied to the encoded data block (step **508**). Upon completion of the encoding of the input data block, an encoded data block with its corresponding desirability factor is output from each (enabled) encoder E**1** . . . En and maintained in a corresponding buffer (step **510**), and the encoded data block size is counted (step **512**).

Next, a compression ratio obtained by each enabled encoder is calculated by taking the ratio of the size of the input data block (as determined by the input counter **10**) to the size of the encoded data block output from each enabled encoder (step **514**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **516**). A determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **518**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **518**), then the original unencoded input data block is selected for output and a null data compression type descriptor (as discussed above) is appended thereto (step **520**). Accordingly, the original unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **522**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **518**), then a figure of merit is calculated for each encoded data block having a compression ratio which exceeds the compression ratio threshold limit (step **524**). Again, the figure of merit for a given encoded data block is comprised of a weighted average of the a priori user specified threshold and the corresponding encoder desirability factor associated with the encoded data block. Next, the encoded data block having the greatest figure of merit is selected for output (step **526**). An appropriate data compression type descriptor is then appended (step **528**) to indicate the data encoding technique applied to the encoded data block. The encoded data block (which has the greatest figure of merit) along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **530**).

After the encoded data block or the unencoded input data block is output (steps **530** and **522**), a determination is made as to whether the input data stream contains additional data blocks to be processed (step **532**). If the input data stream includes additional data blocks (affirmative result in step **532**), then the next successive data block is received (step **534**), its block size is counted (return to step **502**) and the data compression process is iterated for each successive data block in the input data stream. Once the final input data block is processed (negative result in step **532**), data compression of the input data stream is finished (step **536**).

Referring now to FIG. **6**, a block diagram illustrates a data compression system according to another embodiment of the present invention. The data compression system depicted in FIG. **6** is similar to the data compression system discussed in detail above with reference to FIG. **2** except that the embodiment of FIG. **6** includes an a priori specified timer that provides real-time or pseudo real-time output data. In particular, an interval timer **90**, operatively coupled to the encoder module **30**, is preloaded with a user specified time value. The role of the interval timer (as will be explained in greater detail below with reference to FIGS. **7***a* and **7***b*) is to limit the processing time for each input data block processed by the encoder module **30** so as to ensure that the real-time, pseudo real-time, or other time critical nature of the data compression processes is preserved.

The operation of the data compression system of FIG. **6** will now be discussed in further detail with reference to the flow diagram of FIGS. **7***a* and **7***b*. A data stream comprising one or more data blocks is input into the data compression system and the first data block in the data stream is received

(step **700**), and its size is determined by the counter module **10** (step **702**). The data block is then stored in buffer **20** (step **704**).

Next, concurrent with the completion of the receipt and counting of the first data block, the interval timer **90** is initialized (step **706**) and starts counting towards a user-specified time limit. The input data block is then sent to the encoder module **30** wherein data compression of the data block by each (enabled) encoder E**1** . . . En commences (step **708**). Next, a determination is made as to whether the user specified time expires before the completion of the encoding process (steps **710** and **712**). If the encoding process is completed before or at the expiration of the timer, i.e., each encoder (E**1** through En) completes its respective encoding process (negative result in step **710** and affirmative result in step **712**), then an encoded data block is output from each (enabled) encoder E**1** . . . En and maintained in a corresponding buffer (step **714**).

On the other hand, if the timer expires (affirmative result in **710**), the encoding process is halted (step **716**). Then, encoded data blocks from only those enabled encoders E**1** . . . En that have completed the encoding process are selected and maintained in buffers (step **718**). It is to be appreciated that it is not necessary (or in some cases desirable) that some or all of the encoders complete the encoding process before the interval timer expires. Specifically, due to encoder data dependency and natural variation, it is possible that certain encoders may not operate quickly enough and, therefore, do not comply with the timing constraints of the end use. Accordingly, the time limit ensures that the real-time or pseudo real-time nature of the data encoding is preserved.

After the encoded data blocks are buffered (step **714** or **718**), the size of each encoded data block is counted (step **720**). Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10**) to the size of the encoded data block output from each enabled encoder (step **722**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **724**). A determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **726**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **726**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **728**). The original unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **730**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **726**), then the encoded data block having the greatest compression ratio is selected (step **732**). An appropriate data compression type descriptor is then appended (step **734**). The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **736**).

After the encoded data block or the unencoded input data block is output (steps **730** or **736**), a determination is made as to whether the input data stream contains additional data blocks to be processed (step **738**). If the input data stream includes additional data blocks (affirmative result in step **738**), the next successive data block is received (step **740**), its block size is counted (return to step **702**) and the data compression process in repeated. This process is iterated for each data block in the input data stream, with each data block being processed within the user-specified time limit as discussed above. Once the final input data block is processed (negative result in step **738**), data compression of the input data stream is complete (step **742**).

Referring now to FIG. **8**, a block diagram illustrates a content independent data compression system according to another embodiment of the present system. The data compression system of FIG. **8** incorporates all of the features discussed above in connection with the system embodiments of FIGS. **2**, **4**, and **6**. For example, the system of FIG. **8** incorporates both the a priori specified timer for providing real-time or pseudo real-time of output data, as well as the enhanced metric for selecting an optimal encoding technique. Based on the foregoing discussion, the operation of the system of FIG. **8** is understood by those skilled in the art.

Referring now to FIG. **9**, a block diagram illustrates a data compression system according to a preferred embodiment of the present invention. The system of FIG. **9** contains many of the features of the previous embodiments discussed above. However, this embodiment advantageously includes a cascaded encoder module **30***c* having an encoding architecture comprising a plurality of sets of serially-cascaded encoders Em,n, where "m" refers to the encoding path (i.e., the encoder set) and where "n" refers to the number of encoders in the respective path. It is to be understood that each set of serially cascaded encoders can include any number of disparate and/or similar encoders (i.e., n can be any value for a given path m).

The system of FIG. **9** also includes a output buffer module **40***c* which comprises a plurality of buffer/counters B/C m,n, each associated with a corresponding one of the encoders Em,n. In this embodiment, an input data block is sequentially applied to successive encoders (encoder stages) in the encoder path so as to increase the data compression ratio. For example, the output data block from a first encoder E**1**,**1**, is buffered and counted in B/C**1**,**1**, for subsequent processing by a second encoder E**1**,**2**. Advantageously, these parallel sets of sequential encoders are applied to the input data stream to effect content free lossless data compression. This embodiment provides for multi-stage sequential encoding of data with the maximum number of encoding steps subject to the available real-time, pseudo real-time, or other timing constraints.

As with each previously discussed embodiment, the encoders Em,n may include those lossless encoding techniques currently well known within the art, including: run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. Encoding techniques are selected based upon their ability to effectively encode different types of input data. A full complement of encoders provides for broad coverage of existing and future data types. The input data blocks may be applied simultaneously to the encoder paths (i.e., the encoder paths may operate in parallel, utilizing task multiplexing on a single central processor, or via dedicated hardware, or by executing on a plurality of processor or dedicated hardware systems, or any combination thereof). In addition, an input data block may be sequentially applied to the encoder paths. Moreover, each serially cascaded encoder path may comprise a fixed (predetermined) sequence of encoders or a random sequence of encoders. Advantageously, by simultaneously or sequentially processing input data blocks via a plurality of sets of serially cascaded encoders, content free data compression is achieved.

The operation of the data compression system of FIG. **9** will now be discussed in further detail with reference to the flow diagram of FIGS. **10***a* and **10***b*. A data stream comprising one or more data blocks is input into the data compression system and the first data block in the data stream is received (step **100**), and its size is determined by the counter module **10** (step **102**). The data block is then stored in buffer **20** (step **104**).

Next, concurrent with the completion of the receipt and counting of the first data block, the interval timer **90** is initialized (step **106**) and starts counting towards a user-specified time limit. The input data block is then sent to the cascade encoder module **30**C wherein the input data block is applied to the first encoder (i.e., first encoding stage) in each of the cascaded encoder paths E**1**,**1** . . . Em,**1** (step **108**). Next, a determination is made as to whether the user specified time expires before the completion of the first stage encoding process (steps **110** and **112**). If the first stage encoding process is completed before the expiration of the timer, i.e., each encoder (E**1**,**1** . . . Em,**1**) completes its respective encoding process (negative result in step **110** and affirmative result in step **112**), then an encoded data block is output from each encoder E**1**,**1** . . . Em,**1** and maintained in a corresponding buffer (step **114**). Then for each cascade encoder path, the output of the completed encoding stage is applied to the next successive encoding stage in the cascade path (step **116**). This process (steps **110**, **112**, **114**, and **116**) is repeated until the earlier of the timer expiration (affirmative result in step **110**) or the completion of encoding by each encoder stage in the serially cascaded paths, at which time the encoding process is halted (step **118**).

Then, for each cascade encoder path, the buffered encoded data block output by the last encoder stage that completes the encoding process before the expiration of the timer is selected for further processing (step **120**). Advantageously, the interim stages of the multi-stage data encoding process are preserved. For example, the results of encoder E**1**,**1** are preserved even after encoder E**1**,**2** begins encoding the output of encoder E**1**,**1**. If the interval timer expires after encoder E**1**,**1** completes its respective encoding process but before encoder E**1**,**2** completes its respective encoding process, the encoded data block from encoder E**1**,**1** is complete and is utilized for calculating the compression ratio for the corresponding encoder path. The incomplete encoded data block from encoder E**1**,**2** is either discarded or ignored.

It is to be appreciated that it is not necessary (or in some cases desirable) that some or all of the encoders in the cascade encoder paths complete the encoding process before the interval timer expires. Specifically, due to encoder data dependency, natural variation and the sequential application of the cascaded encoders, it is possible that certain encoders may not operate quickly enough and therefore do not comply with the timing constraints of the end use. Accordingly, the time limit ensures that the real-time or pseudo real-time nature of the data encoding is preserved.

After the encoded data blocks are selected (step **120**), the size of each encoded data block is counted (step **122**). Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10**) to the size of the encoded data block output from each encoder (step **124**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **126**). A determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **128**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **128**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **130**). The original unencoded data block and its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **132**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **128**), then a figure of merit is calculated for each encoded data block having a compression ratio which exceeds the compression ratio threshold limit (step **134**). Again, the figure of merit for a given encoded data block is comprised of a weighted average of the a priori user specified threshold and the corresponding encoder desirability factor associated with the encoded data block. Next, the encoded data block having the greatest figure of merit is selected (step **136**). An appropriate data compression type descriptor is then appended (step **138**) to indicate the data encoding technique applied to the encoded data block. For instance, the data type compression descriptor can indicate that the encoded data block was processed by either a single encoding type, a plurality of sequential encoding types, and a plurality of random encoding types. The encoded data block (which has the greatest figure of merit) along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **140**).

After the unencoded data block or the encoded data input data block is output (steps **132** and **140**), a determination is made as to whether the input data stream contains additional data blocks to be processed (step **142**). If the input data stream includes additional data blocks (affirmative result in step **142**), then the next successive data block is received (step **144**), its block size is counted (return to step **102**) and the data compression process is iterated for each successive data block in the input data stream. Once the final input data block is processed (negative result in step **142**), data compression of the input data stream is finished (step **146**).

Referring now to FIG. **11**, a block diagram illustrates a data decompression system according to one embodiment of the present invention. The data decompression system preferably includes an input buffer **1100** that receives as input an uncompressed or compressed data stream comprising one or more data blocks. The data blocks may range in size from individual bits through complete files or collections of multiple files. Additionally, the data block size may be fixed or variable. The input data buffer **1100** is preferably included (not required) to provide storage of input data for various hardware implementations. A descriptor extraction module **1102** receives the buffered (or unbuffered) input data block and then parses, lexically, syntactically, or otherwise analyzes the input data block using methods known by those skilled in the art to extract the data compression type descriptor associated with the data block. The data compression type descriptor may possess values corresponding to null (no encoding applied), a single applied encoding technique, or multiple encoding techniques applied in a specific or random order (in accordance with the data compression system embodiments and methods discussed above).

A decoder module **1104** includes a plurality of decoders D**1** . . . Dn for decoding the input data block using a decoder, set of decoders, or a sequential set of decoders corresponding to the extracted compression type descriptor. The decoders D**1** . . . Dn may include those lossless encoding techniques currently well known within the art, including: run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. Decod-

ing techniques are selected based upon their ability to effectively decode the various different types of encoded input data generated by the data compression systems described above or originating from any other desired source. As with the data compression systems discussed above, the decoder module **1104** may include multiple decoders of the same type applied in parallel so as to reduce the data decoding time.

The data decompression system also includes an output data buffer **1106** for buffering the decoded data block output from the decoder module **1104**.

The operation of the data decompression system of FIG. **11** will be discussed in further detail with reference to the flow diagram of FIG. **12**. A data stream comprising one or more data blocks of compressed or uncompressed data is input into the data decompression system and the first data block in the stream is received (step **1200**) and maintained in the buffer (step **1202**). As with the data compression systems discussed above, data decompression is performed on a per data block basis. The data compression type descriptor is then extracted from the input data block (step **1204**). A determination is then made as to whether the data compression type descriptor is null (step **1206**). If the data compression type descriptor is determined to be null (affirmative result in step **1206**), then no decoding is applied to the input data block and the original undecoded data block is output (or maintained in the output buffer) (step **1208**).

On the other hand, if the data compression type descriptor is determined to be any value other than null (negative result in step **1206**), the corresponding decoder or decoders are then selected (step **1210**) from the available set of decoders D**1** . . . Dn in the decoding module **1104**. It is to be understood that the data compression type descriptor may mandate the application of: a single specific decoder, an ordered sequence of specific decoders, a random order of specific decoders, a class or family of decoders, a mandatory or optional application of parallel decoders, or any combination or permutation thereof. The input data block is then decoded using the selected decoders (step **1212**), and output (or maintained in the output buffer **1106**) for subsequent data processing, storage, or transmittal (step **1214**). A determination is then made as to whether the input data stream contains additional data blocks to be processed (step **1216**). If the input data stream includes additional data blocks (affirmative result in step **1216**), the next successive data block is received (step **1220**), and buffered (return to step **1202**). Thereafter, the data decompression process is iterated for each data block in the input data stream. Once the final input data block is processed (negative result in step **1216**), data decompression of the input data stream is finished (step **1218**).

In other embodiments of the present invention described below, data compression is achieved using a combination of content dependent data compression and content independent data compression. For example, FIGS. **13***a* and **13***b* are block diagrams illustrating a data compression system employing both content independent and content dependent data compression according to one embodiment of the present invention, wherein content independent data compression is applied to a data block when the content of the data block cannot be identified or is not associable with a specific data compression algorithm. The data compression system comprises a counter module **10** that receives as input an uncompressed or compressed data stream. It is to be understood that the system processes the input data stream in data blocks that may range in size from individual bits through complete files or collections of multiple files. Additionally, the data block size may be fixed or variable. The counter module **10** counts the size of each input data block (i.e., the data block size is counted in bits, bytes, words, any convenient data multiple or metric, or any combination thereof).

An input data buffer **20**, operatively connected to the counter module **10**, may be provided for buffering the input data stream in order to output an uncompressed data stream in the event that, as discussed in further detail below, every encoder fails to achieve a level of compression that exceeds a priori specified content independent or content dependent minimum compression ratio thresholds. It is to be understood that the input data buffer **20** is not required for implementing the present invention.

A content dependent data recognition module **1300** analyzes the incoming data stream to recognize data types, data strictures, data block formats, file substructures, file types, and/or any other parameters that may be indicative of either the data type/content of a given data block or the appropriate data compression algorithm or algorithms (in serial or in parallel) to be applied. Optionally, a data file recognition list(s) or algorithm(s) **1310** module may be employed to hold and/or determine associations between recognized data parameters and appropriate algorithms. Each data block that is recognized by the content data compression module **1300** is routed to a content dependent encoder module **1320**, if not the data is routed to the content independent encoder module **30**.

A content dependent encoder module **1320** is operatively connected to the content dependent data recognition module **1300** and comprises a set of encoders D**1**, D**2**, D**3** . . . Dm. The encoder set D**1**, D**2**, D**3** . . . Dm may include any number "n" of those lossless or lossy encoding techniques currently well known within the art such as MPEG4, various voice codecs, MPEG3, AC**3**, AAC, as well as lossless algorithms such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. It is to be understood that the encoding techniques are selected based upon their ability to effectively encode different types of input data. It is to be appreciated that a full complement of encoders and or codecs are preferably selected to provide a broad coverage of existing and future data types.

The content independent encoder module **30**, which is operatively connected to the content dependent data recognition module **1300**, comprises a set of encoders E**1**, E**2**, E**3** . . . En. The encoder set E**1**, E**2**, E**3** . . . En may include any number "n" of those lossless encoding techniques currently well known within the art such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. Again, it is to be understood that the encoding techniques are selected based upon their ability to effectively encode different types of input data. It is to be appreciated that a full complement of encoders are preferably selected to provide a broad coverage of existing and future data types.

The encoder modules (content dependent **1320** and content independent **30**) selectively receive the buffered input data blocks (or unbuffered input data blocks from the counter module **10**) from module **1300** based on the results of recognition. Data compression is performed by the respective encoder modules wherein some or all of the encoders D**1** . . . Dm or E**1** . . . En processes a given input data block and outputs a corresponding set of encoded data blocks. It is to be appreciated that the system affords a user the option to enable/disable any one or more of the encoders D**1** . . . Dm and E**1** . . . En prior to operation. As is understood by those skilled in the art, such feature allows the user to tailor the operation of the data compression system for specific applications. It is to be further appreciated that the encoding process may be performed either in parallel or sequentially. In particular, the

encoder set D**1** through Dm of encoder module **1320** and/or the encoder set E**1** through En of encoder module **30** may operate in parallel (i.e., simultaneously processing a given input data block by utilizing task multiplexing on a single central processor, via dedicated hardware, by executing on a plurality of processor or dedicated hardware systems, or any combination thereof). In addition, encoders D**1** through Dm and E**1** through En may operate sequentially on a given unbuffered or buffered input data block. This process is intended to eliminate the complexity and additional processing overhead associated with multiplexing concurrent encoding techniques on a single central processor and/or dedicated hardware, set of central processors and/or dedicated hardware, or any achievable combination. It is to be further appreciated that encoders of the identical type may be applied in parallel to enhance encoding speed. For instance, encoder E**1** may comprise two parallel Huffman encoders for parallel processing of an input data block. It should be further noted that one or more algorithms may be implemented in dedicated hardware such as an MPEG4 or MP3 encoding integrated circuit.

Buffer/counter modules **1330** and **40** are operatively connected to their respective encoding modules **1320** and **30**, for buffering and counting the size of each of the encoded data blocks output from the respective encoder modules. Specifically, the content dependent buffer/counter **1330** comprises a plurality of buffer/counters BCD**1**, BCD**2**, BCD**3** . . . BCDm, each operatively associated with a corresponding one of the encoders D**1** . . . Dm. Similarly the content independent buffer/counters BCE**1**, BCE**2**, BCE**3** . . . BCEn, each operatively associated with a corresponding one of the encoders E**1** . . . En. A compression ratio module **1340**, operatively connected to the content dependent output buffer/counters **1330** and content independent buffer/counters **40** determines the compression ratio obtained for each of the enabled encoders D**1** . . . Dm and or E**1** . . . En by taking the ratio of the size of the input data block to the size of the output data block stored in the corresponding buffer/counters BCD**1**, BCD**2**, BCD**3** . . . BCDm and or BCE**1**, BCE**2**, BCE**3** . . . BCEn. In addition, the compression ratio module **1340** compares each compression ratio with an a priori-specified compression ratio threshold limit to determine if at least one of the encoded data blocks output from the enabled encoders BCD**1**, BCD**2**, BCD**3** . . . BCDm and or BCE**1**, BCE**2**, BCE**3** . . . BCEn achieves a compression that meets an a priori-specified threshold. As is understood by those skilled in the art, the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It should be noted that different threshold values may be applied to content dependent and content independent encoded data. Further these thresholds may be adaptively modified based upon enabled encoders in either or both the content dependent or content independent encoder sets, along with any associated parameters. A compression type description module **1350**, operatively coupled to the compression ratio module **1340**, appends a corresponding compression type descriptor to each encoded data block which is selected for output so as to indicate the type of compression format of the encoded data block.

A mode of operation of the data compression system of FIGS. **13***a* and **13***b* will now be discussed with reference to the flow diagrams of FIGS. **14***a*-**14***d*, which illustrates a method for performing data compression using a combination of content dependent and content independent data compression. In general, content independent data compression is applied to a given data block when the content of a data block cannot be identified or is not associated with a specific data compression algorithm. More specifically, referring to FIG. **14***a*, a data stream comprising one or more data blocks is input into the data compression system and the first data block in the stream is received (step **1400**). As stated above, data compression is performed on a per data block basis. As previously stated a data block may represent any quantity of data from a single bit through a multiplicity of files or packets and may vary from block to block. Accordingly, the **10** first input data block in the input data stream is input into the counter module **10** that counts the size of the data block (step **1402**). The data block is then stored in the buffer **20** (step **1404**). The data block is then analyzed on a per block or multi-block basis by the content dependent data recognition module **1300** (step **1406**). If the data stream content is not recognized utilizing the recognition list(s) or algorithms(s) module **1310** (step **1408**) the data is routed to the content independent encoder module **30** and compressed by each (enabled) encoder E**1** . . . En (step **1410**). Upon completion of the encoding of the input data block, an encoded data block is output from each (enabled) encoder E**1** . . . En and maintained in a corresponding buffer (step **1412**), and the encoded data block size is counted (step **1414**).

Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10** to the size of each encoded data block output from the enabled encoders (step **1416**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **1418**). It is to be understood that the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It is to be further understood that notwithstanding that the current limit for lossless data compression is the entropy limit (the present definition of information content) for the data, the present invention does not preclude the use of future developments in lossless data compression that may increase lossless data compression ratios beyond what is currently known within the art. Additionally the content independent data compression threshold may be different from the content dependent threshold and either may be modified by the specific enabled encoders.

After the compression ratios are compared with the threshold, a determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **1420**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **1420**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **1434**). A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the input data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1436**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **1420**), then the encoded data block having the greatest compression ratio is selected (step **1422**). An appropriate data compression type descriptor is then appended (step **1424**). A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to

enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1426**).

As previously stated the data block stored in the buffer **20** (step **1404**) is analyzed on a per block or multi-block basis by the content dependent data recognition module **1300** (step **1406**). If the data stream content is recognized utilizing the recognition list(s) or algorithms(s) module **1310** (step **1434**) the appropriate content dependent algorithms are enabled and initialized (step **1436**), and the data is routed to the content dependent encoder module **1320** and compressed by each (enabled) encoder D**1** . . . Dm (step **1438**). Upon completion of the encoding of the input data block, an encoded data block is output from each (enabled) encoder D**1** . . . Dm and maintained in a corresponding buffer (step **1440**), and the encoded data block size is counted (step **1442**).

Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10** to the size of each encoded data block output from the enabled encoders (step **1444**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **1448**). It is to be understood that the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It is to be further understood that many of these algorithms may be lossy, and as such the limits may be subject to or modified by an end target storage, listening, or viewing device. Further notwithstanding that the current limit for lossless data compression is the entropy limit (the present definition of information content) for the data, the present invention does not preclude the use of future developments in lossless data compression that may increase lossless data compression ratios beyond what is currently known within the art. Additionally the content independent data compression threshold may be different from the content dependent threshold and either may be modified by the specific enabled encoders.

After the compression ratios are compared with the threshold, a determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **1420**). If there are no encoded datablocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **1420**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **1434**). A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the input data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1436**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **1420**), then the encoded data block having the greatest compression ratio is selected (step **1422**). An appropriate data compression type descriptor is then appended (step **1424**). A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1426**).

After the encoded data block or the unencoded data input data block is output (steps **1426** and **1436**), a determination is made as to whether the input data stream contains additional data blocks to be processed (step **1428**). If the input data stream includes additional data blocks (affirmative result in step **1428**), the next successive data block is received (step **1432**), its block size is counted (return to step **1402**) and the data compression process in repeated. This process is iterated for each data block in the input data stream. Once the final input data block is processed (negative result in step **1428**), data compression of the input data stream is finished (step **1430**).

Since a multitude of data types may be present within a given input data block, it is often difficult and/or impractical to predict the level of compression that will be achieved by a specific encoder. Consequently, by processing the input data blocks with a plurality of encoding techniques and comparing the compression results, content free data compression is advantageously achieved. Further the encoding may be lossy or lossless dependent upon the input data types. Further if the data type is not recognized the default content independent lossless compression is applied. It is not a requirement that this process be deterministic—in fact a certain probability may be applied if occasional data loss is permitted. It is to be appreciated that this approach is scalable through future generations of processors, dedicated hardware, and software. As processing capacity increases and costs reduce, the benefits provided by the present invention will continue to increase. It should again be noted that the present invention may employ any lossless data encoding technique.

FIGS. **15***a* and **15***b* are block diagrams illustrating a data compression system employing both content independent and content dependent data compression according to another embodiment of the present invention. The system in FIGS. **15***a* and **15***b* is similar in operation to the system of FIGS. **13***a* and **13***b* in that content independent data compression is applied to a data block when the content of the data block cannot be identified or is not associable with a specific data compression algorithm. The system of FIGS. **15***a* and **15***b* additionally performs content independent data compression on a data block when the compression ratio obtained for the data block using the content dependent data compression does not meet a specified threshold.

A mode of operation of the data compression system of FIGS. **15***a* and **15***b* will now be discussed with reference to the flow diagram of FIGS. **16***a*-**16***d*, which illustrates a method for performing data compression using a combination of content dependent and content independent data compression. A data stream comprising one or more data blocks is input into the data compression system and the first data block in the stream is received (step **1600**). As stated above, data compression is performed on a per data block basis. As previously stated a data block may represent any quantity of data from a single bit through a multiplicity of files or packets and may vary from block to block. Accordingly, the first input data block in the input data stream is input into the counter module **10** that counts the size of the data block (step **1602**). The data block is then stored in the buffer **20** (step **1604**). The data block is then analyzed on a per block or multi-block basis by the content dependent data recognition module **1300** (step **1606**). If the data stream content is not recognized utilizing

the recognition list(s) or algorithms(s) module **1310** (Step **1608**) the data is routed to the content independent encoder module **30** and compressed by each (enabled) encoder E**1** . . . En (step **1610**). Upon completion of the encoding of the input data block, an encoded data block is output from each (enabled) encoder E**1** . . . En and maintained in a corresponding buffer (step **1612**), and the encoded data block size is counted (step **1614**).

Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10** to the size of each encoded data block output from the enabled encoders (step **1616**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **1618**). It is to be understood that the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It is to be further understood that notwithstanding that the current limit for lossless data compression is the entropy limit (the present definition of information content) for the data, the present invention does not preclude the use of future developments in lossless data compression that may increase lossless data compression ratios beyond what is currently known within the art. Additionally the content independent data compression threshold may be different from the content dependent threshold and either may be modified by the specific enabled encoders.

After the compression ratios are compared with the threshold, a determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **1620**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **1620**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **1634**). A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the input data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1636**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **1620**), then the encoded data block having the greatest compression ratio is selected (step **1622**). An appropriate data compression type descriptor is then appended (step **1624**). A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1626**).

As previously stated the data block stored in the buffer **20** (step **1604**) is analyzed on a per block or multi-block basis by the content dependent data recognition module **1300** (step **1606**). If the data stream content is recognized utilizing the recognition list(s) or algorithms(s) module **1310** (step **1634**) the appropriate content dependent algorithms are enabled and initialized (step **1636**) and the data is routed to the content dependent encoder module **1620** and compressed by each (enabled) encoder D**1** . . . Dm (step **1638**). Upon completion of the encoding of the input data block, an encoded data block is output from each (enabled) encoder D**1** . . . Dm and maintained in a corresponding buffer (step **1640**), and the encoded data block size is counted (step **1642**).

Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10** to the size of each encoded data block output from the enabled encoders (step **1644**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **1648**). It is to be understood that the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It is to be further understood that many of these algorithms may be lossy, and as such the limits may be subject to or modified by an end target storage, listening, or viewing device. Further notwithstanding that the current limit for lossless data compression is the entropy limit (the present definition of information content) for the data, the present invention does not preclude the use of future developments in lossless data compression that may increase lossless data compression ratios beyond what is currently known within the art. Additionally the content independent data compression threshold may be different from the content dependent threshold and either may be modified by the specific enabled encoders.

After the compression ratios are compared with the threshold, a determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **1648**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **1620**), then the original unencoded input data block is routed to the content independent encoder module **30** and the process resumes with compression utilizing content independent encoders (step **1610**).

After the encoded data block or the unencoded data input data block is output (steps **1626** and **1636**), a determination is made as to whether the input data stream contains additional data blocks to be processed (step **1628**). If the input data stream includes additional data blocks (affirmative result in step **1628**), the next successive data block is received (step **1632**), its block size is counted (return to step **1602**) and the data compression process in repeated. This process is iterated for each data block in the input data stream. Once the final input data block is processed (negative result in step **1628**), data compression of the input data stream is finished (step **1630**).

FIGS. **17***a* and **17***b* are block diagrams illustrating a data compression system employing both content independent and content dependent data compression according to another embodiment of the present invention. The system in FIGS. **17***a* and **17***b* is similar in operation to the system of FIGS. **13***a* and **13***b* in that content independent data compression is applied to a data block when the content of the data block cannot be identified or is not associable with a specific data compression algorithm. The system of FIGS. **17***a* and **17***b* additionally uses a priori estimation algorithms or look-up tables to estimate the desirability of using content independent data compression encoders and/or content dependent data compression encoders and selecting appropriate algorithms or subsets thereof based on such estimation.

More specifically, a content dependent data recognition and or estimation module **1700** is utilized to analyze the incoming data stream for recognition of data types, data strictures, data block formats, file substructures, file types, or any other parameters that may be indicative of the appropriate

data compression algorithm or algorithms (in serial or in parallel) to be applied. Optionally, a data file recognition list(s) or algorithm(s) **1710** module may be employed to hold associations between recognized data parameters and appropriate algorithms. If the content data compression module recognizes a portion of the data, that portion is routed to the content dependent encoder module **1320**, if not the data is routed to the content independent encoder module **30**. It is to be appreciated that process of recognition (modules **1700** and **1710**) is not limited to a deterministic recognition, but may further comprise a probabilistic estimation of which encoders to select for compression from the set of encoders of the content dependent module **1320** or the content independent module **30**. For example, a method may be employed to compute statistics of a data block whereby a determination that the locality of repetition of characters in a data stream is determined is high can suggest a text document, which may be beneficially compressed with a lossless dictionary type algorithm. Further the statistics of repeated characters and relative frequencies may suggest a specific type of dictionary algorithm. Long strings will require a wide dictionary file while a wide diversity of strings may suggest a deep dictionary. Statistics may also be utilized in algorithms such as Huffman where various character statistics will dictate the choice of different Huffman compression tables. This technique is not limited to lossless algorithms but may be widely employed with lossy algorithms. Header information in frames for video files can imply a specific data resolution. The estimator then may select the appropriate lossy compression algorithm and compression parameters (amount of resolution desired). As shown in previous embodiments of the present invention, desirability of various algorithms and now associated resolutions with lossy type algorithms may also be applied in the estimation selection process.

A mode of operation of the data compression system of FIGS. **17***a* and **17***b* will now be discussed with reference to the flow diagrams of FIGS. **18***a*-**18***d*. The method of FIGS. **18***a*-**18***d* use a priori estimation algorithms or look-up tables to estimate the desirability or probability of using content independent data compression encoders or content dependent data compression encoders, and select appropriate or desirable algorithms or subsets thereof based on such estimates. A data stream comprising one or more data blocks is input into the data compression system and the first data block in the stream is received (step **1800**). As stated above, data compression is performed on a per data block basis. As previously stated a data block may represent any quantity of data from a single bit through a multiplicity of files or packets and may vary from block to block. Accordingly, the first input data block in the input data stream is input into the counter module **10** that counts the size of the data block (step **1802**). The data block is then stored in the buffer **20** (step **1804**). The data block is then analyzed on a per block or multi-block basis by the content dependent/content independent data recognition module **1700** (step **1806**). If the data stream content is not recognized utilizing the recognition list(s) or algorithms(s) module **1710** (step **1808**) the data is to the content independent encoder module **30**. An estimate of the best content independent encoders is performed (step **1850**) and the appropriate encoders are enabled and initialized as applicable. The data is then compressed by each (enabled) encoder E**1** . . . En (step **1810**). Upon completion of the encoding of the input data block, an encoded data block is output from each (enabled) encoder E**1** . . . En and maintained in a corresponding buffer (step **1812**), and the encoded data block size is counted (step **1814**).

Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10** to the size of each encoded data block output from the enabled encoders (step **1816**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **1818**). It is to be understood that the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It is to be further understood that notwithstanding that the current limit for lossless data-compression is the entropy limit (the present definition of information content) for the data, the present invention does not preclude the use of future developments in lossless data compression that may increase lossless data compression ratios beyond what is currently known within the art. Additionally the content independent data compression threshold may be different from the content dependent threshold and either may be modified by the specific enabled encoders.

After the compression ratios are compared with the threshold, a determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **1820**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **1820**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **1834**). A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the input data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1836**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **1820**), then the encoded data block having the greatest compression ratio is selected (step **1822**). An appropriate data compression type descriptor is then appended (step **1824**). A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1826**).

As previously stated the data block stored in the buffer **20** (step **1804**) is analyzed on a per block or multi-block basis by the content dependent data recognition module **1300** (step **1806**). If the data stream content is recognized or estimated utilizing the recognition list(s) or algorithms(s) module **1710** (affirmative result in step **1808**) the recognized data type/file or block is selected based on a list or algorithm (step **1838**) and an estimate of the desirability of using the associated content dependent algorithms can be determined (step **1840**). For instance, even though a recognized data type may be associated with three different encoders, an estimation of the desirability of using each encoder may result in only one or two of the encoders being actually selected for use. The data is routed to the content dependent encoder module **1320** and compressed by each (enabled) encoder D**1** . . . Dm (step **1842**). Upon completion of the encoding of the input data

block, an encoded data block is output from each (enabled) encoder D1 . . . Dm and maintained in a corresponding buffer (step **1844**), and the encoded data block size is counted (step **1846**).

Next, a compression ratio is calculated for each encoded data block by taking the ratio of the size of the input data block (as determined by the input counter **10** to the size of each encoded data block output from the enabled encoders (step **1848**). Each compression ratio is then compared with an a priori-specified compression ratio threshold (step **1850**). It is to be understood that the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. It is to be further understood that many of these algorithms may be lossy, and as such the limits may be subject to or modified by an end target storage, listening, or viewing device. Further notwithstanding that the current limit for lossless data compression is the entropy limit (the present definition of information content) for the data, the present invention does not preclude the use of future developments in lossless data compression that may increase lossless data compression ratios beyond what is currently known within the art. Additionally the content independent data compression threshold may be different from the content dependent threshold and either may be modified by the specific enabled encoders.

After the compression ratios are compared with the threshold, a determination is made as to whether the compression ratio of at least one of the encoded data blocks exceeds the threshold limit (step **1820**). If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit (negative determination in step **1820**), then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto (step **1834**). A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the input data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1836**).

On the other hand, if one or more of the encoded data blocks possess a compression ratio greater than the compression ratio threshold limit (affirmative result in step **1820**), then the encoded data block having the greatest compression ratio is selected (step **1822**). An appropriate data compression type descriptor is then appended (step **1824**). A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing, storage, or transmittal (step **1826**).

After the encoded data block or the unencoded data input data block is output (steps **1826** and **1836**), a determination is made as to whether the input data stream contains additional data blocks to be processed (step **1828**). If the input data stream includes additional data blocks (affirmative result in step **1428**), the next successive data block is received (step **1832**), its block size is counted (return to step **1802**) and the data compression process in repeated. This process is iterated for each data block in the input data stream. Once the final

input data block is processed (negative result in step **1828**), data compression of the input data stream is finished (step **1830**).

It is to be appreciated that in the embodiments described above with reference to FIGS. **13-18**, an a priori specified time limit or any other real-time requirement may be employed to achieve practical and efficient real-time operation.

Although illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present invention is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention. All such Changes and modifications are intended to be included within the scope of the invention as defined by the appended claims.

What is claimed is:

1. A method of decompressing one or more compressed data packets of a data stream using a data decompression processor, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to a data packet, the method comprising:

receiving a data packet from the data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders used to compress data blocks associated with the data packet, and further wherein the lossless encoders are selected based on analyses of content of the data blocks;

analyzing the data packet to identify a descriptor;

selecting one or more lossless decoders for a data block associated with the data packet, wherein the selecting is based on the descriptor;

decompressing the data block with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data block is encoded utilizing content dependent data compression; and

decompressing the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression.

2. The method of claim **1**, wherein the descriptor comprises values corresponding to a single applied decoding technique or multiple decoding techniques applied in a specific order.

3. The method of claim **1**, wherein the step of decoding the data block utilizing content independent data decompression occurs prior to the step of decoding the data block utilizing content dependent data decompression.

4. The method of claim **1**, wherein the multiple decoders are applied to decompress a plurality of data blocks associated with the data packet of the data stream.

5. The method of claim **4**, wherein a plurality of data packets of the data stream are decompressed.

6. The method of claim **5**, further comprising generating a decompressed data stream from outputs of the multiple decoders.

7. The method of claim **1**, wherein the method of decompressing the one or more compressed data packets of the data stream is performed in real-time.

8. A system for decompressing one or more compressed data packets of a data stream, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to a data packet, the system comprising:

an input interface that receives a data packet from the data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors

indicate lossless encoders used to compress data blocks associated with the data packet, and further wherein the lossless encoders are selected based on analyses of content of the data blocks;

a data decompression processor operatively connected to said input interface having a computer readable program code of instructions executable by the data decompression processor, said instructions comprising:

analyze the data packet to identify a descriptor;

select one or more lossless decoders for a data block associated with the data packet, wherein the selecting is based on the descriptor;

decompress the data block with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data block is encoded utilizing content dependent data compression;

decompress the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression; and

an output interface operatively connected to said data decompression processor that outputs a decompressed data packet.

9. The system of claim **8**, wherein the descriptor comprises values corresponding to a single applied decoding technique or multiple decoding techniques applied in a specific order.

10. The system of claim **8**, wherein the instruction to decode the data block utilizing content independent data decompression is performed prior to the instruction to decode the data block utilizing content dependent data decompression.

11. The system of claim **8**, wherein the multiple decoders are applied to decompress a plurality of data blocks associated with the data packet of the data stream.

12. The system of claim **11**, wherein the system for decompressing one or more compressed data packets of a data stream operates on a plurality of data packets of the data stream.

13. The system of claim of **8**, wherein the system for decompressing one or more compressed data packets of a data stream operates in real-time.

14. A method of compressing a plurality of data blocks to create a compressed data packet in a data stream using a data compression processor, wherein multiple encoders applying a plurality of lossless compression techniques are applied to data blocks, the method comprising:

receiving a data block;

analyzing content of the data block to determine a data block type;

selecting one or more lossless encoders based on the data block type and a computer file, wherein the computer file indicates data block types and associated lossless encoders;

compressing the data block with a selected encoder utilizing content dependent data compression, if the data block type is recognized as associated with a lossless encoder utilizing content dependent data compression;

compressing the data block with a selected lossless encoder utilizing content independent data compression, if the data block type is not recognized as associated with a lossless encoder utilizing content dependent data compression; and

providing a descriptor for the compressed data packet in the data stream, wherein the descriptor indicates the one or more selected lossless encoders for the encoded data block.

15. The method of claim **14**, wherein the multiple encoders are applied to compress a plurality of data blocks associated with the compressed data packet of the data stream.

16. The method of claim **15**, wherein a plurality of data packets of the data stream are compressed.

17. The method of claim **16**, further comprising generating a compressed data stream from outputs of the multiple encoders.

18. The method of claim **14**, wherein the method of compressing a plurality of data blocks is performed in real-time.

19. A system for compressing a plurality of data blocks to create a compressed data packet in a data stream, wherein multiple encoders applying a plurality of lossless compression techniques are applied to data blocks, comprising:

an input interface that receives a data block;

a data compression processor operatively connected to said input interface having a computer readable program code of instructions executable by the data compression processor, said instructions comprising:

analyze content of the data block to determine a data block type;

select one or more lossless encoders based on the data block type and a computer file, wherein the computer file indicates data block types and associated lossless encoders;

compress the data block with a selected lossless encoder utilizing content dependent data compression, if the data block type is recognized as associated with a lossless encoder utilizing content dependent data compression;

compress the data block with a selected lossless encoder utilizing content independent data compression, if the data block type is not recognized as associated with a lossless encoder utilizing content dependent data compression; and

an output interface operatively connected to said data compression processor that outputs a descriptor comprising one or more values in the compressed data packet in the data stream, wherein the descriptor indicates the selected one or more lossless encoders.

20. The system of claim **19**, wherein the multiple decoders are applied to compress a plurality of data blocks associated with the compressed data packet of the data stream.

21. The system of claim **20**, wherein the system for compressing outputs a plurality of data packets in the data stream.

22. The system of claim **19**, wherein the system for compressing operates in real-time.

* * * * *

US007777651B2

(12) United States Patent
Fallon et al.

(10) **Patent No.: US 7,777,651 B2**
(45) **Date of Patent: *Aug. 17, 2010**

(54) **SYSTEM AND METHOD FOR DATA FEED ACCELERATION AND ENCRYPTION**

(75) Inventors: **James J. Fallon**, Armonk, NY (US); **Paul F. Pickel**, Bethpage, NY (US); **Stephen J. McErlain**, New York, NY (US); **Carlton W. Melone**, Ridgewood, NJ (US)

(73) Assignee: **Realtime Data LLC**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 324 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/131,631**

(22) Filed: **Jun. 2, 2008**

(65) **Prior Publication Data**

US 2009/0287839 A1 Nov. 19, 2009

Related U.S. Application Data

(63) Continuation of application No. 10/434,305, filed on May 7, 2003, now Pat. No. 7,417,568, which is a continuation-in-part of application No. 09/969,987, filed on Oct. 3, 2001.

(60) Provisional application No. 60/378,517, filed on May 7, 2002, provisional application No. 60/237,571, filed on Oct. 3, 2000.

(51) **Int. Cl.**
H03M 7/34 (2006.01)

(52) **U.S. Cl.** **341/51**; 341/50; 341/67; 341/75; 341/79

(58) **Field of Classification Search** 341/50, 341/51, 67, 79, 75
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,394,352	A	7/1968	Wernikoff et al.
3,490,690	A	1/1970	Apple et al.
4,021,782	A	5/1977	Hoerning
4,032,893	A	6/1977	Moran
4,054,951	A	10/1977	Jackson et al.
4,302,775	A	11/1981	Widergren et al.
4,325,085	A	4/1982	Gooch
4,360,840	A	11/1982	Wolfrun et al.

(Continued)

FOREIGN PATENT DOCUMENTS

DE 4127518 2/1992

(Continued)

OTHER PUBLICATIONS

Expand Networks Accelerator 4000 Series User's Guide, 1999.

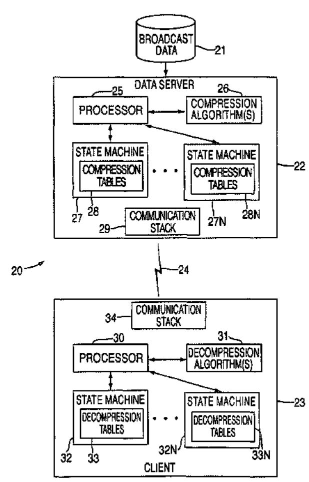
(Continued)

Primary Examiner—Jean B Jeanglaude
(74) *Attorney, Agent, or Firm*—Sterne, Kessler, Goldstein & Fox PLLC

(57) **ABSTRACT**

The transmission of broadcast data, such as financial data and news feeds, is accelerated over a communication channel using data compression and decompression to provide secure transmission and transparent multiplication of communication bandwidth, as well as reduce latency. Broadcast data may include packets having fields. Encoders associated with particular fields may be selected to compress those particular fields.

123 Claims, 6 Drawing Sheets



U.S. PATENT DOCUMENTS

4,386,416	A	5/1983	Giltner et al.
4,394,774	A	7/1983	Widergren et al.
4,494,108	A	1/1985	Langdon, Jr. et al.
4,499,499	A	2/1985	Brickman et al.
4,574,351	A	3/1986	Dang et al.
4,593,324	A	6/1986	Ohkubo et al.
4,626,829	A	12/1986	Hauck
4,682,150	A	7/1987	Mathes et al.
4,701,745	A	10/1987	Waterworth
4,729,020	A	3/1988	Schaphorst et al.
4,730,348	A	3/1988	MacCrisken
4,748,638	A	5/1988	Friedman et al.
4,754,351	A	6/1988	Wright
4,804,959	A	2/1989	Makansi et al.
4,813,040	A	3/1989	Futato
4,862,167	A	8/1989	Copeland, III
4,866,601	A	9/1989	DuLac et al.
4,870,415	A	9/1989	Van Maren et al.
4,872,009	A	10/1989	Tsukiyama et al.
4,876,541	A	10/1989	Storer
4,888,812	A	12/1989	Dinan et al.
4,890,282	A	12/1989	Lambert et al.
4,897,717	A	1/1990	Hamilton et al.
4,906,991	A	3/1990	Fiala et al.
4,906,995	A	3/1990	Swanson
4,929,946	A	5/1990	O'Brien et al.
4,953,324	A	9/1990	Herrmann
4,956,808	A	9/1990	Aakre et al.
4,965,675	A	10/1990	Hori et al.
4,988,998	A	1/1991	O'Brien
5,003,307	A	3/1991	Whiting et al.
5,016,009	A	5/1991	Whiting et al.
5,028,922	A	7/1991	Huang
5,045,848	A	9/1991	Fascenda
5,045,852	A	9/1991	Mitchell et al.
5,046,027	A	9/1991	Taaffe et al.
5,049,881	A	9/1991	Gibson et al.
5,091,782	A	2/1992	Krause et al.
5,097,261	A	3/1992	Langdon, Jr. et al.
5,109,226	A	4/1992	Mac Lean, Jr. et al.
5,109,433	A	4/1992	Notenboom
5,113,522	A	5/1992	Dinwiddie, Jr. et al.
5,115,309	A	5/1992	Hang
5,121,342	A	6/1992	Szymborski
5,126,739	A	6/1992	Whiting et al.
5,128,963	A	7/1992	Akagiri
5,146,221	A	9/1992	Whiting et al.
5,150,430	A	9/1992	Chu
5,155,484	A	10/1992	Chambers, IV
5,159,336	A	10/1992	Rabin et al.
5,167,034	A	11/1992	MacLean, Jr. et al.
5,175,543	A	12/1992	Lantz
5,179,651	A	1/1993	Taaffe et al.
5,187,793	A	2/1993	Keith et al.
5,191,431	A	3/1993	Hasegawa et al.
5,204,756	A	4/1993	Chevion et al.
5,209,220	A	5/1993	Hiyama et al.
5,212,742	A	5/1993	Normile et al.
5,226,176	A	7/1993	Westaway et al.
5,227,893	A	7/1993	Ett
5,231,492	A	7/1993	Dangi et al.
5,237,460	A	8/1993	Miller et al.
5,237,675	A	8/1993	Hannon, Jr.
5,243,341	A	9/1993	Seroussi et al.
5,243,348	A	9/1993	Jackson
5,247,638	A	9/1993	O'Brien et al.
5,247,646	A	9/1993	Osterlund et al.
5,249,053	A	9/1993	Jain
5,263,168	A	11/1993	Toms et al.
5,270,832	A	12/1993	Balkanski et al.
5,280,600	A	1/1994	Van Maren et al.
5,287,420	A	2/1994	Barrett
5,289,580	A	2/1994	Latif et al.
5,293,379	A	3/1994	Carr
5,293,576	A	3/1994	Mihm, Jr. et al.
5,307,497	A	4/1994	Feigenbaum et al.
5,309,555	A	5/1994	Akins et al.
5,341,440	A	8/1994	Earl et al.
5,347,600	A	9/1994	Barnsley et al.
5,355,498	A	10/1994	Provino et al.
5,357,614	A	10/1994	Pattisam et al.
5,367,629	A	11/1994	Chu et al.
5,373,290	A	12/1994	Lempel et al.
5,379,036	A	1/1995	Storer
5,379,757	A	1/1995	Hiyama et al.
5,381,145	A	1/1995	Allen et al.
5,389,922	A	2/1995	Seroussi et al.
5,394,534	A	2/1995	Kulakowski et al.
5,396,228	A	3/1995	Garahi
5,400,401	A	3/1995	Wasilewski et al.
5,403,639	A	4/1995	Belsan et al.
5,406,278	A	4/1995	Graybill et al.
5,406,279	A	4/1995	Anderson et al.
5,410,671	A	4/1995	Elgamal et al.
5,412,384	A	5/1995	Chang et al.
5,414,850	A	5/1995	Whiting
5,420,639	A	5/1995	Perkins
5,434,983	A	7/1995	Yaso et al.
5,437,020	A	7/1995	Wells et al.
5,452,287	A	9/1995	Dicecco
5,454,079	A	9/1995	Roper et al.
5,454,107	A	9/1995	Lehman et al.
5,455,576	A	10/1995	Clark, II et al.
5,455,680	A	10/1995	Shin
5,461,679	A	10/1995	Normile et al.
5,463,390	A	10/1995	Whiting et al.
5,467,087	A	11/1995	Chu
5,471,206	A	11/1995	Allen et al.
5,479,587	A	12/1995	Campbell et al.
5,479,633	A	12/1995	Wells et al.
5,483,470	A	1/1996	Alur et al.
5,486,826	A	1/1996	Remillard
5,488,364	A	1/1996	Cole
5,495,244	A	2/1996	Jeong et al.
5,504,842	A	4/1996	Gentile
5,506,844	A	4/1996	Rao
5,506,872	A	4/1996	Mohler
5,506,944	A	4/1996	Gentile
5,528,628	A	6/1996	Park et al.
5,530,845	A	6/1996	Hiatt et al.
5,533,051	A	7/1996	James
5,535,311	A	7/1996	Zimmerman
5,535,356	A	7/1996	Kim et al.
5,535,369	A	7/1996	Wells et al.
5,537,658	A	7/1996	Bakke et al.
5,539,865	A	7/1996	Gentile
5,542,031	A	7/1996	Douglass et al.
5,544,290	A	8/1996	Gentile
5,546,395	A	8/1996	Sharma et al.
5,546,475	A	8/1996	Bolle et al.
5,553,160	A	9/1996	Dawson
5,557,551	A	9/1996	Craft
5,557,668	A	9/1996	Brady
5,557,749	A	9/1996	Norris
5,561,421	A	10/1996	Smith et al.
5,561,824	A	10/1996	Carreiro et al.
5,563,961	A	10/1996	Rynderman et al.
5,574,952	A	11/1996	Brady et al.
5,574,953	A	11/1996	Rust et al.
5,576,953	A	11/1996	Hugentobler
5,581,715	A	12/1996	Verinsky et al.
5,583,500	A	12/1996	Allen et al.
5,586,264	A	12/1996	Belknap et al.
5,586,285	A	12/1996	Hasbun et al.

5,590,306	A	12/1996	Watanabe et al.
5,596,674	A	1/1997	Bhandari et al.
5,604,824	A	2/1997	Chui et al.
5,606,706	A	2/1997	Takamoto et al.
5,610,657	A	3/1997	Zhang
5,611,024	A	3/1997	Campbell et al.
5,612,788	A	3/1997	Stone
5,613,069	A	3/1997	Walker
5,615,017	A	3/1997	Choi
5,615,287	A	3/1997	Fu et al.
5,619,995	A	4/1997	Lobodzinski
5,621,820	A	4/1997	Rynderman et al.
5,623,623	A	4/1997	Kim et al.
5,623,701	A	4/1997	Bakke et al.
5,627,534	A	5/1997	Craft
5,627,995	A	5/1997	Miller et al.
5,629,732	A	5/1997	Moskowitz et al.
5,630,092	A	5/1997	Carreiro et al.
5,635,632	A	6/1997	Fay et al.
5,635,932	A	6/1997	Shinagawa et al.
5,638,498	A	6/1997	Tyler et al.
5,640,158	A	6/1997	Okayama et al.
5,642,506	A	6/1997	Lee
5,649,032	A	7/1997	Burt et al.
5,652,795	A	7/1997	Dillon et al.
5,652,857	A	7/1997	Shimoi et al.
5,652,917	A	7/1997	Maupin et al.
5,654,703	A	8/1997	Clark, II
5,655,138	A	8/1997	Kikinis
5,666,560	A	9/1997	Moertl et al.
5,668,737	A	9/1997	Iler
5,671,355	A	9/1997	Collins
5,671,389	A	9/1997	Saliba
5,671,413	A	9/1997	Shipman et al.
5,675,333	A	10/1997	Boursier et al.
5,675,789	A	10/1997	Ishii et al.
5,686,916	A	11/1997	Bakhmutsky
5,692,159	A	11/1997	Shand
5,694,619	A	12/1997	Konno
5,696,927	A	12/1997	MacDonald et al.
5,703,793	A	12/1997	Wise et al.
5,708,511	A	1/1998	Gandhi et al.
5,715,477	A	2/1998	Kikinis
5,717,393	A	2/1998	Nakano et al.
5,717,394	A	2/1998	Schwartz et al.
5,719,862	A	2/1998	Lee et al.
5,721,958	A	2/1998	Kikinis
5,724,475	A	3/1998	Kirsten
5,729,228	A	3/1998	Franaszek et al.
5,740,395	A	4/1998	Wells et al.
5,742,773	A	4/1998	Blomfield-Brown et al.
5,748,904	A	5/1998	Huang et al.
5,757,852	A	5/1998	Jericevic et al.
5,765,027	A	6/1998	Wang et al.
5,767,898	A	6/1998	Urano et al.
5,768,445	A	6/1998	Troeller et al.
5,768,525	A	6/1998	Kralowetz et al.
5,771,340	A	6/1998	Nakazato et al.
5,774,715	A	6/1998	Madany et al.
5,778,411	A	7/1998	DeMoss et al.
5,781,767	A	7/1998	Inoue et al.
5,784,572	A	7/1998	Rostoker et al.
5,787,487	A	7/1998	Hashimoto et al.
5,794,229	A	8/1998	French et al.
5,796,864	A	8/1998	Callahan
5,799,110	A	8/1998	Israelsen et al.
5,805,834	A	9/1998	McKinley et al.
5,805,932	A	9/1998	Kawashima et al.
5,808,660	A	9/1998	Sekine et al.
5,809,176	A	9/1998	Yajima
5,809,299	A	9/1998	Cloutier et al.
5,809,337	A	9/1998	Hannah et al.
5,812,195	A	9/1998	Zhang
5,812,789	A	9/1998	Diaz et al.
5,818,368	A	10/1998	Langley
5,818,369	A	10/1998	Withers
5,818,530	A	10/1998	Canfield et al.
5,819,215	A	10/1998	Dobson et al.
5,822,781	A	10/1998	Wells et al.
5,825,424	A	10/1998	Canfield et al.
5,825,830	A	10/1998	Kopf
5,832,037	A	11/1998	Park
5,832,126	A	11/1998	Tanaka
5,835,788	A	11/1998	Blumer et al.
5,836,003	A	11/1998	Sadeh
5,838,996	A	11/1998	deCarmo
5,839,100	A	11/1998	Wegener
5,841,979	A	11/1998	Schulhof et al.
5,847,762	A	12/1998	Canfield
5,850,565	A	12/1998	Wightman
5,861,824	A	1/1999	Ryu et al.
5,861,920	A	1/1999	Mead et al.
5,864,342	A	1/1999	Kajiya et al.
5,867,167	A	2/1999	Deering
5,867,602	A	2/1999	Zandi et al.
5,870,036	A	2/1999	Franaszek et al.
5,870,087	A	2/1999	Chau
5,872,530	A	2/1999	Domyo et al.
5,874,907	A	2/1999	Craft
5,883,975	A	3/1999	Narita et al.
5,884,269	A	3/1999	Cellier et al.
5,886,655	A	3/1999	Rust
5,887,165	A	3/1999	Martel et al.
5,889,961	A	3/1999	Dobbek
5,892,847	A	4/1999	Johnson
5,909,557	A	6/1999	Betker et al.
5,909,559	A	6/1999	So
5,915,079	A	6/1999	Vondran, Jr. et al.
5,917,438	A	6/1999	Ando
5,918,068	A	6/1999	Shafe'
5,918,225	A	6/1999	White et al.
5,920,326	A	7/1999	Rentschler et al.
5,923,860	A	7/1999	Olarig
5,930,358	A	7/1999	Rao
5,936,616	A	8/1999	Torborg, Jr. et al.
5,943,692	A	8/1999	Marberg et al.
5,945,933	A	8/1999	Kalkstein
5,949,355	A	9/1999	Panaoussis
5,951,623	A	9/1999	Reynar et al.
5,955,976	A	9/1999	Heath
5,956,490	A	9/1999	Buchholz et al.
5,960,465	A	9/1999	Adams
5,964,842	A	10/1999	Packard
5,968,149	A	10/1999	Jaquette et al.
5,969,927	A	10/1999	Schirmer et al.
5,973,630	A	10/1999	Heath
5,974,235	A	10/1999	Nunally et al.
5,974,387	A	10/1999	Kageyama
5,974,471	A	10/1999	Belt
5,978,483	A	11/1999	Thompson, Jr. et al.
5,982,360	A	11/1999	Wu et al.
5,982,723	A	11/1999	Kamatani
5,987,022	A	11/1999	Geiger
5,987,590	A	11/1999	Wing So
5,990,884	A	11/1999	Douma et al.
5,991,515	A	11/1999	Fall et al.
5,996,033	A	11/1999	Chiu-Hao
6,000,009	A	12/1999	Brady
6,002,411	A	12/1999	Dye
6,003,115	A	12/1999	Spear et al.
6,008,743	A	12/1999	Jaquette
6,011,901	A	1/2000	Kirsten
6,014,694	A	1/2000	Aharoni et al.
6,023,755	A	2/2000	Casselman
6,026,217	A	2/2000	Adiletta
6,028,725	A	2/2000	Blumenau

6,031,939	A	2/2000	Gilbert et al.	
6,032,148	A	2/2000	Wilkes	
6,058,459	A	5/2000	Owen et al.	
6,061,398	A	5/2000	Satoh et al.	
6,070,179	A	5/2000	Craft	
6,073,232	A	6/2000	Kroeker et al.	
6,075,470	A	6/2000	Little et al.	
6,078,958	A	6/2000	Echeita et al.	
6,091,777	A	7/2000	Guetz et al.	
6,092,123	A	7/2000	Steffan et al.	
6,094,634	A	7/2000	Yahagi et al.	
6,097,520	A	8/2000	Kadnier	
6,098,114	A	8/2000	McDonald et al.	
6,104,389	A	8/2000	Ando	
6,105,130	A	8/2000	Wu et al.	
6,128,412	A	10/2000	Satoh	
6,134,631	A	10/2000	Jennings, III	
6,141,053	A	10/2000	Saukkonen	
6,145,020	A	11/2000	Barnett	
6,145,069	A	11/2000	Dye	
6,169,241	B1	1/2001	Shimizu	
6,170,007	B1	1/2001	Venkatraman et al.	
6,170,047	B1	1/2001	Dye	
6,170,049	B1	1/2001	So	
6,172,936	B1	1/2001	Kitazaki	
6,173,381	B1	1/2001	Dye	
6,175,650	B1	1/2001	Sindhu	
6,175,856	B1	1/2001	Riddle	
6,182,125	B1	1/2001	Borella et al.	
6,185,625	B1	2/2001	Tso et al.	
6,185,659	B1	2/2001	Milillo et al.	
6,192,082	B1	2/2001	Moriarty et al.	
6,192,155	B1	2/2001	Fan	
6,195,024	B1	2/2001	Fallon	
6,195,465	B1	2/2001	Zandi et al.	
6,208,273	B1	3/2001	Dye et al.	
6,215,904	B1	4/2001	Lavallee	
6,219,754	B1	4/2001	Belt et al.	
6,222,886	B1	4/2001	Yogeshwar	
6,225,922	B1	5/2001	Norton	
6,226,667	B1	5/2001	Matthews et al.	
6,226,740	B1	5/2001	Iga	
6,230,223	B1	5/2001	Olarig	
6,237,054	B1	5/2001	Freitag, Jr.	
6,243,829	B1	6/2001	Chan	
6,253,264	B1	6/2001	Sebastian	
6,272,178	B1 *	8/2001	Nieweglowski et al.	375/240.03
6,272,627	B1	8/2001	Mann	
6,272,628	B1	8/2001	Aguilar et al.	
6,282,641	B1	8/2001	Christensen	
6,298,408	B1	10/2001	Park	
6,308,311	B1	10/2001	Carmichael et al.	
6,309,424	B1	10/2001	Fallon	
6,310,563	B1	10/2001	Har et al.	
6,317,714	B1	11/2001	Del Castillo et al.	
6,330,622	B1	12/2001	Schaefer	
6,345,307	B1	2/2002	Booth	
6,356,589	B1	3/2002	Gebler et al.	
6,356,937	B1	3/2002	Montville et al.	
6,392,567	B2	5/2002	Satoh	
6,404,931	B1	6/2002	Chen et al.	
6,421,387	B1	7/2002	Rhee	
6,434,168	B1	8/2002	Kari	
6,434,695	B1	8/2002	Esfahani et al.	
6,442,659	B1	8/2002	Blumenau	
6,449,658	B1	9/2002	Lafe et al.	
6,449,682	B1	9/2002	Toorians	
6,452,602	B1	9/2002	Morein	
6,452,933	B1	9/2002	Duffield et al.	
6,459,429	B1	10/2002	Deering	
6,463,509	B1	10/2002	Teoman et al.	
6,487,640	B1	11/2002	Lipasti	
6,489,902	B2	12/2002	Heath	
6,513,113	B1	1/2003	Kobayashi	
6,523,102	B1	2/2003	Dye et al.	
6,526,174	B1	2/2003	Graffagnino	
6,529,633	B1	3/2003	Easwar et al.	
6,532,121	B1	3/2003	Rust et al.	
6,539,438	B1	3/2003	Ledzius et al.	
6,539,456	B2	3/2003	Stewart	
6,542,644	B1	4/2003	Satoh	
6,577,254	B2	6/2003	Rasmussen	
6,590,609	B1	7/2003	Kitade et al.	
6,597,812	B1	7/2003	Fallon et al.	
6,601,104	B1	7/2003	Fallon	
6,604,040	B2	8/2003	Kawasaki et al.	
6,604,158	B1	8/2003	Fallon	
6,606,040	B2 *	8/2003	Abdat	341/87
6,606,413	B1	8/2003	Zeineh	
6,609,223	B1	8/2003	Wolfgang	
6,618,728	B1	9/2003	Rail	
6,624,761	B2	9/2003	Fallon	
6,650,261	B2	11/2003	Nelson et al.	
6,661,839	B1	12/2003	Ishida et al.	
6,661,845	B1	12/2003	Herath	
6,704,840	B2	3/2004	Nalawadi et al.	
6,711,709	B1	3/2004	York	
6,717,534	B2	4/2004	Yokose	
6,731,814	B2	5/2004	Zeck et al.	
6,745,282	B2	6/2004	Okada et al.	
6,748,457	B2	6/2004	Fallon	
6,756,922	B2	6/2004	Ossia	
6,768,749	B1	7/2004	Osler et al.	
6,792,151	B1	9/2004	Barnes et al.	
6,810,434	B2	10/2004	Muthujumaraswathy et al.	
6,813,689	B2	11/2004	Baxter, III	
6,819,271	B2	11/2004	Geiger et al.	
6,822,589	B1	11/2004	Dye et al.	
6,856,651	B2	2/2005	Singh	
6,862,278	B1	3/2005	Chang et al.	
6,879,266	B1	4/2005	Dye et al.	
6,885,316	B2	4/2005	Mehring	
6,885,319	B2	4/2005	Geiger et al.	
6,888,893	B2	5/2005	Li et al.	
6,909,383	B2	6/2005	Shokrollahi et al.	
6,944,740	B2	9/2005	Abali et al.	
6,959,359	B1	10/2005	Suzuki et al.	
6,963,608	B1	11/2005	Wu	
6,993,597	B2	1/2006	Nakagawa et al.	
7,007,099	B1	2/2006	Donati et al.	
7,054,493	B2	5/2006	Schwartz	
7,069,342	B1	6/2006	Biederman	
7,089,391	B2	8/2006	Geiger et al.	
7,102,544	B1	9/2006	Liu	
7,129,860	B2	10/2006	Alvarez, II et al.	
7,130,913	B2	10/2006	Fallon	
7,161,506	B2	1/2007	Fallon	
7,181,608	B2	2/2007	Fallon et al.	
7,190,284	B1	3/2007	Dye et al.	
7,319,667	B1	1/2008	Biederman	
7,321,937	B2	1/2008	Fallon	
7,330,912	B1	2/2008	Fox et al.	
7,352,300	B2	4/2008	Fallon	
7,358,867	B2	4/2008	Fallon	
7,376,772	B2	5/2008	Fallon	
7,378,992	B2	5/2008	Fallon	
7,386,046	B2	6/2008	Fallon et al.	
7,395,345	B2	7/2008	Fallon	
7,400,274	B2	7/2008	Fallon et al.	
7,415,530	B2	8/2008	Fallon	
7,417,568	B2	8/2008	Fallon et al.	
7,552,069	B2	6/2009	Kepecs	
7,565,441	B2	7/2009	Romanik et al.	
7,714,747	B2 *	5/2010	Fallon	341/51
2001/0031092	A1	10/2001	Zeck et al.	
2001/0032128	A1	10/2001	Kepecs	

2001/0047473	A1	11/2001	Fallon
2001/0052038	A1	12/2001	Fallon et al.
2001/0054131	A1	12/2001	Alvarez, II et al.
2002/0037035	A1	3/2002	Singh
2002/0069354	A1	6/2002	Fallon et al.
2002/0080871	A1	6/2002	Fallon et al.
2002/0097172	A1	7/2002	Fallon
2002/0101367	A1	8/2002	Geiger et al.
2002/0104891	A1	8/2002	Otto
2002/0126755	A1	9/2002	Li et al.
2002/0191692	A1	12/2002	Fallon et al.
2003/0030575	A1	2/2003	Frachtenberg et al.
2003/0034905	A1	2/2003	Anton et al.
2003/0084238	A1	5/2003	Okada et al.
2003/0090397	A1	5/2003	Rausmussen
2003/0142874	A1	7/2003	Schwartz
2003/0191876	A1	10/2003	Fallon
2004/0042506	A1	3/2004	Fallon et al.
2004/0056783	A1	3/2004	Fallon
2004/0073710	A1	4/2004	Fallon
2004/0073746	A1	4/2004	Fallon
2006/0015650	A1	1/2006	Fallon
2006/0181441	A1	8/2006	Fallon
2006/0181442	A1	8/2006	Fallon
2006/0184687	A1	8/2006	Fallon
2006/0184696	A1	8/2006	Fallon
2006/0190644	A1	8/2006	Fallon
2006/0195601	A1	8/2006	Fallon
2007/0043939	A1	2/2007	Fallon et al.
2007/0050514	A1	3/2007	Fallon
2007/0050515	A1	3/2007	Fallon
2007/0067483	A1	3/2007	Fallon
2007/0083746	A1	4/2007	Fallon et al.
2007/0109154	A1	5/2007	Fallon
2007/0109155	A1	5/2007	Fallon
2007/0109156	A1	5/2007	Fallon
2007/0174209	A1	7/2007	Fallon
2008/0232457	A1	9/2008	Fallon et al.
2009/0154545	A1	6/2009	Fallon et al.

FOREIGN PATENT DOCUMENTS

EP	0164677	12/1985
EP	0185098	6/1986
EP	0283798	9/1988
EP	0595406	5/1994
EP	0405572	11/1994
EP	0718751	6/1996
EP	0493130	6/1997
EP	0587437	2/2002
GB	2162025	1/1986
JP	6051989	2/1994
JP	9188009	7/1997
JP	11149376	6/1999
WO	WO 9414273	6/1994
WO	WO 9429852	12/1994
WO	WO 9502873	1/1995
WO	WO 9748212	6/1997
WO	WO 9908186	2/1999
WO	WO 02/39591	5/2002

OTHER PUBLICATIONS

Tridgell, Andrew; “Efficient Algorithms for Sorting and Synchronization”; A thesis submitted for the degree of Doctor of Philosophy at The Australian National University; Feb. 1999; pp. iii-106.
Jung, et al.; “Performance optimization of wireless local area networks through VLSI data compression”; Wireless Networks, vol. 4, 1998; pp. 27-39.
Jones, et al.; “Lossless data compression for short duration 3D frames in positron emission tomography”, IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference; vol. 3; pp. 1831-1834.
Maier, Mark W.; “Algorithm Evaluation for the Synchronous Data Compression Standard”; University of Alabama; pp. 1-10.
Bassiouni, et al.; “A Scheme for Data Compression in Supercomputers”; IEEE; 1988; pp. 272-278.
Welch, Terry A.; “A Technique for High-Performance Data Compression”; IEEE; Jun. 1984; pp. 8-19.
ALDC: Adaptive Lossless Data Compression; IBM; 1994.
ALDC-Macro: Adaptive Lossless Data Compression; IBM Corporation; 1994.
ALDC1-20S: Adaptive Lossless Data Compression; IBM Corporation; 1994.
ALDC1-40S: Adaptive Lossless Data Compression; IBM Corporation; 1994.
ALDC1-5S: Adaptive Lossless Data Compression; IBM Corporation; 1994.
Craft, David J.; “Data Compression Choice No Easy Call”; Computer Technology Review; vol. XIV, No. 1; Jan. 1994.
Costlow, Terry; “Sony designs faster, denser tape drive”; Electronic Engineering Times; May 20, 1996, pp. 86-87.
Wilson, Ron; “IBM ups compression ante”; Electronic Engineering Times; Aug. 16, 1993; pp. 1-94.
“IBM Announces New Feature for 3480 Subsystem”; Tucson Today; vol. 12, No. 337, Jul. 25, 1989.
Syngress Media, Inc.; “CCA Citrix Certified Administrator for MetaFrame 1.8 Study Guide”; 2000.
International Telecommunication Union; “Data Compression Procedures for Data Circuit Terminating Equipment (DCE) Using Error Correction Procedures”; Geneva, 1990.
Cheng, et al.; “A fast, highly reliable data compression chip and algorithm for storage systems”; IBM J. Res. Develop.; vol. 40, No. 6, Nov. 1996; pp. 603-613.
Cisco Systems; “Cisco IOS Data Compression”; 1997; pp. 1-10.
Craft, D. J.; “A fast hardware data compression algorithm and some algorithmic extensions”; IBM J. Res. Develop.; vol. 42; No. 6; Nov. 6, 1998; pp. 733-746.
Rustici, Robert; “Enhanced CU-SeeMe”1995, Zero In Technologies, Inc.
White Pine Software; “CU-SeeMe Pro: Quick Start Guide”; Version 4.0 for Windows; 1999.
“CU-SeeMe Reflector”; www.geektimes.com/michael/CU-SeeMe/faqs/reflectors.html; accessed on Dec. 2, 2008.
Daniels, et al.; “Citrix WinFrame 1.6 Beta”; May 1, 1996; license.icopyright.net/user/downloadLicense.act?lic=3.7009-9123; accessed Dec. 2, 2008.
Held, et al.; “Data Compression”; Third Edition; John Wiley & Sons Ltd.; 1991.
Data Compression Applications and Innovations Workshop; Proceedings of a Workshop held in Conjunction with the IEEE Data Compression Conference; Snowbird, Utah; Mar. 31, 1995.
Britton, et al.; “Discovery Desktop Conferencing with NetMeeting 2.0”; IDG Books Worldwide, inc.; 1997.
Sattler, Michael; “Internet TV with CU-SeeMe”; Sams.Net Publishing; 1995; First Edition.
IBM Microelectronics Comdex Fall ’93 Booth Location.
Disz, et al.; “Performance Model of the Argonne Voyager Multimedia Server”; IEEE; 1997; pp. 316-327.
“Downloading and Installing NetMeeting”; www.w4mq.com/help/h3.htm; accessed on Dec. 2, 2008.
Fox, et al.; “Adapting to Network and Client Variability via On-Demand Dynamic Distillation”; ASPLOS VII; Oct. 1996; pp. 160-170.
Fox, et al.; “Adapting to Network and Client Variation Using Infrastructural Proxies: Lessons and Perceptives”; IEEE Personal Communications, Aug. 1998; pp. 10-19.
Han, et al.; “CU-SeeMe VR Immersive Desktop Teleconferencing”; Department of Computer Science; Cornell University; To appear in ACM Multimedia 1996.
Howard, et al.; “Parallel Lossless Image Compression Using Huffman and Arithmetic Coding”; 1992; pp. 1-9.
Howard, Paul G.; “Text Image Compression Using Soft Pattern Matching”; The Computer Journal; vol. 40, No. 2/3; 1997; pp. 146-156.

Howard, et al.; "The Emerging JBIG2 Standard"; IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, No. 7, Nov. 1998; pp. 838-848.
Craft, D. J.; "A fast hardware data compression algorithm and some algorithmic extensions"; Journal of Research and Development; vol. 42, No. 6, Nov. 1998; pp. 733-745.
"Direct Access Storage Device Compression and Decompression Data Flow"; IBM Technical Disclosure Bulletin; vol. 38, No. 11; Nov. 1995; pp. 291-295.
ICA Timeline, Sep. 24, 2007.
Converse, et al.; "Low Bandwidth X Extension"; Protocol Version 1.0; X Consortium; Dec. 21, 1996.
Magstar and IBM 3590 High Performance Tape Subsystem Technical Guide; Nov. 1996; IBM International Technical Support Organization.
MetaFrame Administration Student Workbook; Jun. 1998; Citrix Professional Courseware; Citrix Systems, Inc.
NCD WinCenter 3.1: Bringing Windows to Every Desktop; 1998.
Overview NetMeeting 2.1; Microsoft TechNet; technet.microsoft.com/en-us/library/cc767141(printer).aspx; accessed Dec. 2, 2008.
NetMeeting 2.1 Resource Kit; Microsoft TechNet; technet.microsoft.com/en-us/library/cc767142(printer).aspx; accessed on Dec. 2, 2008.
Conferencing Standards: NetMeeeting 2.1 Resource Kit; Microsoft TechNet; technet.microsoft.com/en-us/library/cc767150(printer).aspx; accessed Dec. 2, 2008.
Summers, Bob; "Official Microsoft NetMeeting Book"; Microsoft Press; 1998.
Zebrose, Katherine L.; "Integrating Hardware Accelerators into Internetworking Switches"; Telco Systems.
Simpson, et al.; "A Multiple Processor Approach to Data Compression"; ACM; 1998; pp. 641-649.
"IBM Technology Products Introduces New Family of High-Performance Data Compression Products"; IBM; Aug. 16, 1993.
ReadMe; PowerQuest Drive Image Pro; Version 3.00; 1994-1999; PowerQuest Corporation; p. 1-6.
Schulzrinne, et al.; "RTP Profile for Audio and Video Conferences with Minimal Control"; Jan. 1996; www.ietf.org/rfc/rfc1890.txt; accessed on Dec. 3, 2008.
Zhu, C.; "RTP Payload Format for H.263 Video Streams"; Standards Track; Sep. 1997; pp. 1-12.
Simpson, W.; "The Point-To-Point Protocol (PPP)"; Standards Track; Jul. 1994; pp. i-52.
Reynolds, et al.; "Assigned Numbers"; Standards Track; Oct. 1994; pp. 1-230.
Deutsch, et al.; "ZLIB Compressed Data Format Specification version 3.3"; Informational; May 1996; p. 1-10.
Deutsch, P.; "DEFLATE Compressed Data Format Specification version 1.3"; Informational; May 1996; pp. 1-15.
Rand, D.; "The PPP Compression Control Protocol (CCP)"; Standards Track; Jun. 1996; pp. 1-9.
Schneider, et al.; "PPP LZS-DCP Compression Protocol (LZS-DCP)"; Informational; Aug. 1996; pp. 1-18.
Friend, et al.; "PPP Stac LZS Compression Protocol"; Informational; Aug. 1996; pp. 1-20.
Schneider, et al.; "PPP for Data Compression in Data Circuit-Terminating Equipment (DCE)"; Informational; Aug. 1996; pp. 1-10.
Atkins, et al.; "PGP Message Exchange Formats"; Informational; Aug. 1996; pp. 1-21.
Castineyra, et al.; "The Nimrod Routing Architecture"; Informational, Aug. 1996; pp. 1-27.
Freed, et al.; "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures"; Best Current Practice; Nov. 1996; pp. 1-21.
Shacham, et al.; "IP Payload Compression Protocol (IPComp)"; Standards Track; Dec. 1998; pp. 1-10.
Sidewinder 50 Product Manual; Seagate Technology, Inc.; 1997.
IBM RAMAC Virtual Array; IBM; Jul. 1997.
Bruni, et al.; "DB2or OS/390 and Data Compression" IBM Corporation; Nov. 1998.
Smith, Mark; "Thin Client/Server Computing Works"; WindowsITPro; Nov. 1, 1998; pp. 1-13; license.icopyright.net/user/downloadLicense.act?lic=3.7009-8355; accessed Dec. 2, 2008.
International Telecommunication Union; "Information Technology—Digital Compression and Coding of Continuous-Tone Still Images—Requirements and Guidelines"; 1993.
International Telecommunications Union; "Information technology—Lossless and near-lossless compression of continuous-tone still images—Baseline"; 1999.
Davis, Andrew W.; "The Video Answering Machine: Intel ProShare's Next Step"; Advanced Imaging; vol. 12, No. 3; Mar. 1997; pp. 28, 30.
Abbott, III, Walter D.; "A Simple, Low Overhead Data Compression Algorithm for Converting Lossy Compression Processes to Lossless"; Naval Postgraduate School Thesis; Dec. 1993.
Thomborson, Clark; "V.42bis and Other Ziv-Lempel Variants"; IEEE; 1991; p. 460.
Thomborson, Cark; "The V.42bis Standard for Data-Compressing Modems"; IEEE; Oct. 1992; pp. 41-53.
Sun, Andrew; "Using and Managing PPP"; O'Reilly & Associates, Inc.; 1999.
"What is the V42bis Standard?"; www.faqs.org/faqs/compression-faq/part1/section-10.html; accessed on Dec. 2, 2008.
"The WSDC Download Guide: Drive Image Professional for DOS, OS/2, and Windows"; wsdcds01.watson.ibm.com/WSDC.nsf/Guides/Download/Applications-DriveImage.htm; Accessed Nov. 22, 1999.
"The WSDC Download Guide: Drive Image Professional"; wsdcds01.watson.ibm.com/wsdc.nsf/Guides/Download/Applications-DirveImage.htm; accessed on May 3, 2001.
APPNOTE_TXT from pkware.txt; Version 6.3.2; PKWARE Inc., 1989.
CU-SeeMe readme.txt; Dec. 2, 1995.
CU-seeme txt from indstate.txt; README.TXT for CU-SeeMe version 0.90b1; Mar. 23, 1997.
Cuseeme txt 19960221.txt; CUSEEME.TXT; Feb. 21, 1996.
Citrix Technology Guide, 1997.
Lettieri, et al.; "Data Compression in the V.42bis Modems"; pp. 398-403, (1994).
High Performance x2/V.34+/V.42bis 56K BPS Plug & Play External Voice/FAX/Data Modem User's Manual.
H.323 Protocols Suite; www.protocols.com/pbook/h323.htm.
Hoffman, Roy; "Data Compression in Digital Systems"; Chapman & Hall; 1997; Chapter 14, pp. 344-360.
LBX X Consortium Algorithms; rzdocs.uni-hohenheim.de/aix_4.33/ext_doc/usr/share/man/info/en_US/a_doc_lib/x11 . . . ; X11R6 Technical Specifications.
Basics of Images; www.geom.uiuc.edu/events/courses/1996/cmwh/Stills/basics.html, (1996).
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 6,624,761, U.S. Appl. No. 95/000,464, issued Jul. 24, 2009, 29 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 6,624,761, U.S. Appl. No. 95/000,464, issued Dec. 15, 2009, 20 pgs.
Response to Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, U.S. Appl. No. 95/000,466, filed Oct. 13, 2009, 26 pgs.
Response to Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, U.S. Appl. No. 95/000,466, filed Aug. 24, 2009, 39 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, U.S. Appl. No. 95/000,466, issued Jun. 22, 2009, 11 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 7,321,937, U.S. Appl. No. 95/000,466, issued Jun. 22, 2009, 16 pgs.
Official Action Closing Prosecution for Inter Partes Reexamination of U.S. Pat. No. 7,321,937, U.S. Appl. No. 95/000,466, issued Dec. 22, 2009, 12 pgs.
Comments by Third Party Requester to Patent Owner's Response Inter Partes Reexamination of U.S. Patent No. 7,321,937, U.S. Appl. No. 95/000,466, filed Nov. 10, 2009, 30 pgs.
Supplemental Declaration of Professor James A. Storer, Ph.D. under 37 C.F.R. §1.132 in Inter Partes Reexamination of U.S. Patent No. 7,321,937, U.S. Appl. No. 95/000,466, executed on Nov. 10, 2009, 16 pgs.

Examiner Interview Summary in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, U.S. Appl. No. 90/009,428, issued Dec. 3, 2009, 3 pgs.
Response to Office Action in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, U.S. Appl. No. 90/009,428, filed Dec. 28, 2009, 13 pgs.
Non-Final Office Action in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, U.S. Appl. No. 90/009,428, issued Nov. 2, 2009, 13 pgs.
Official Order Granting Request for Ex Parte Reexamination of U.S. Pat. No. 6,601,104, U.S. Appl. No. 90/009,428, issued Jun. 1, 2009, 12 pgs.
Declaration of Dr. George T. Ligler under 37 C.F.R. §1.132 in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, U.S. Appl. No. 90/009,428, executed Dec. 28, 2009 16 pgs.
Supplementary Declaration of Dr. George T. Ligler under 37 C.F.R. §1.132 in Ex Parte Reexamination of U.S. Pat. No. 6,601,104, U.S. Appl. No. 90/009,428, executed Dec. 30, 2009 1 pg.
Declaration of Dr. George T. Ligler under 37 C.F.R. §1.132 in Inter Partes Reexamination of U.S. Pat. No. 7,321,937, U.S. Appl. No. 95/000,466, executed Aug. 24, 2009 16 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 7,161,506, U.S. Appl. No. 95/000,479, issued Aug. 14, 2009, 41 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,161,506, U.S. Appl. No. 95/000,479, issued Dec. 15, 2009, 37 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 7,378,992, U.S. Appl. No. 95/000,478, issued Aug. 13, 2009, 60 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 7,378,992, U.S. Appl. No. 95/000,478, issued Dec. 15, 2009, 27 pgs.
Official Order Granting Request for Inter Partes Reexamination of U.S. Pat. No. 6,604,158 U.S. Appl. No. 95/000,486, issued Aug. 14, 2009, 35 pgs.
Non-Final Office Action in Inter Partes Reexamination of U.S. Pat. No. 6,604,158, U.S. Appl. No. 95/000,486, issued Nov. 12, 2009, 199 pgs.
Expert Report of Dr. James A. Storer on Invalidity filed on behalf of some of the defendants [Includes Appendices—Exhibits A-K (Exhibit A has been redacted pursuant to a protective order)] filed in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, Jun. 10, 2009, 1090 pgs.
Supplementary Expert Report of Dr. James A. Storer on Invalidity filed on behalf of some of the defendants [Includes Appendices—Exhibits 1-8] filed in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, Jun. 19, 2009, 301 pgs.
Expert Report of James B. Gambrell on Inequitable Conduct filed on behalf of some of the defendants [Includes Appendices—Exhibits A-I] filed in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, Jun. 10, 2009, 199 pgs.
Report and Recommendation of United States Magistrate Judge on Motion for Partial Summary Judgment issued on Jun. 23, 2009, in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, 22 pgs.
Order Adopting Report and Recommendation of United States Magistrate Judge, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, Aug. 24, 2009, 2 pgs.
Opinion and Order of United States Magistrate Judge regarding Claim Construction, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, issued Jun. 22, 2009, 75 pgs.
Script for Defendants' Joint Claim Construction Technology Tutorial Presented to the Magistrate Judge in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, no date on document, 95 pgs.
Script for Realtimes' Technology Tutorial Presented to the Magistrate Judge in *Realtime Data, LLC d/b/a/IXO* v. *Packeteer, Inc. et al.*, Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, Mar. 16, 2009, 69 pgs.
Opinion and Order of United States Magistrate Judge regarding Plaintiff's Motion to Strike Unauthorized New Invalidity Theories from Defendant Citrix's Opening and Reply Briefs in Support of its Motion for Summary Judgment of Invalidity, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, issued Dec. 8, 2009, 10 pgs.
Declaration of Patrick Gogerty, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, executed May 8, 2009, 3 pgs.
Defendant Citrix Systems, Inc.'s Notice Pursuant to 35 U.S.C. Section 282 Disclosures, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 7 pgs.
Blue Coat Defendants' Notice Pursuant to 35 U.S.C. Section 282 Disclosures, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 7 pgs.
Expand Networks' 35 U.S.C. Section 282 Disclosures, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 4 pgs.
Expand Networks' 35 U.S.C. Section 282 Disclosures (Amended), *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 5 pgs.
Defendant Citrix Systems, Inc.'s Notice of Obviousness Combinations Pursuant to Court Order, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 11, 2009, 3 pgs.
Order of United States Magistrate Judge regarding Motion to Limit the Number of Prior Art References to be Asserted at Trail, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 21, 2009, 6 pgs.
Expand Defendants' Notice of Obviousness Combinations Pursuant to Court Order, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 22, 2009, 3 pgs.
Blue Coat Systems, Inc. and 7-Eleven, Inc.'s Notice of Obviousness Combinations to be Used at Trial, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Dec. 22, 2009, 30 pgs.
Defendant Citrix Systems, Inc's Notice of Other Prior Art References Within the Scope of the References Discussed at the Dec. 17, 2009 Hearing, *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District ourt for the Eastern District of Texas, No. 6:08cv144, filed Dec. 29, 2009, 6 pgs.
Docket Listing downloaded Mar. 12, 2010 for *Realtime Data, LLC D/B/A Ixo* v. *Packeteer, Inc., et al.*, District Court for the Eastern District of Texas, No. 6:08cv144, filed Apr. 18, 2008, 165 pgs.
Preliminary Data Sheet, 9600 Data Compressor Processor, Hi/fn, 1997-99, HIFN 000001-68, 68 pgs.
Data Sheet, 9751 Data Compression Processor, 1997-99, HIFN 000069-187, 119 pgs.
Signal Termination Guide, Application Note, Hi/fn, 1997-98, HIFN 000188-194, 7 pgs.
How LZS Data Compression Works, Application Note, Hi/fn, 1997-99, HIFN 000195-207, 13 pgs.
Reference Hardware, 9751 Compression Processor, Hi/fn, 1997-99, HIFN 000208-221, 14 pgs.
Using 9751 in Big Endian Systems, Application Note, Hi/fn, 1997-99, HIFN 000222-234, 13 pgs.
Specification Update, 9751 Compression Processor, Hi/fn, 1997-2000, HIFN 000235-245, 11 pgs.
9732AM Product Release, Hi/fn, 1994-99, HIFN 000246-302, 57 pgs.
Data Sheet, 9732A Data Compression Processor, Hi/fn, 1997-99, HIFN 000303-353, 51 pgs.
9711 to 7711 Migration, Application Note, Hi/fn, 1997-99, HIFN 000354-361, 8 pgs.

Specification Update, 9711 Data Compression Processor, Hi/fn, 1997-99, HIFN 000362-370, 9 pgs.
Differences Between the 9710 & 9711 Processors, Application Note, Hi/fn, 1997-99, HIFN 000371-77, 7 pgs.
Specification Update, 9710 Data Compression Processor, Hi/fn, 1997-99, HIFN 000378-388, 11 pgs.
9706/9706A Data Compression Coprocessor Data Sheet, Stac Electronics, 1991-97, HIFN 000389-473, 85 pgs.
9705/9705A Data Compression Coprocessor, Stac Electronics, 1988-96, HIFN 000474-562, 88 pgs.
9705/9705A Data Compression Coprocessor Data Sheet, Stac Electronics, 1988-96, HIFN 000563-649, 87 pgs.
9700/9701 Compression Coprocessors, Hi/fn, 1997, HIFN 000650-702, 53 pgs.
Data Sheet 9610 Data Compression Processor, Hi/fn, 1997-98, HIFN 000703-744, 42 pgs.
Specification Update 9610 Data Compression Processor, Hi/fn, 1997-99, HIFN 000745-751, 7 pgs.
9705 Data Compression Coprocessor, Stac Electronics, 1988-92, HIFN 000752-831, 80 pgs.
9705 Network Software Design Guide, Application Note, Stac Electronics, 1990-91, HIFN 000832-861, 30 pgs.
Data Sheet 9601 Data Compression Processor, Hi/fn, May 21, 1998, HIFN 000862-920, 59 pgs.
7751 Encryption Processor Reference Kit, Hi/fn, Apr. 1999, HIFN 000921-1114, 194 pgs.
Hardware Data Book, Hi/fn, Nov. 1998, HIFN 001115-1430, 316 pgs.
Data Compression Data Book, Hi/fn, Jan. 1999, HIFN 001431-1889, 459 pgs.
Reference Software 7751 Encryption Processor, Hi/fn, Nov. 1998, HIFN 002164-2201, 38 pgs.
Interface Specification for Synergize Encoding/Decoding Program, JPB, Oct. 10, 1997, HIFN 002215-2216, 2 pgs.
Anderson, Chip, Extended Memory Specification Driver, 1998, HIFN 002217-2264, 48 pgs.
Whiting, Doug, LZS Hardware API, Mar. 12, 1993, HIFN 002265-68, 4 pgs.
Whiting, Doug, Encryption in Sequoia, Apr. 28, 1997, HIFN 002309-2313, 5 pgs.
LZS221-C Version 4 Data Compression Software, Data Sheet, Hi/fn, 1994-97, HIFN 002508-2525, 18 pgs.
eXtended Memory Specification (XMS), ver. 2.0, Microsoft, Jul. 19, 1988, HIFN 002670-2683, 14 pgs.
King, Stanley, Just for Your Info—From Microsoft 2, May 4, 1992, HIFN 002684-2710, 27 pgs.
eXtended Memory Specification (XMS), ver. 2.0, Microsoft, Jul. 19, 1988, HIFN 002711-2724, 14 pgs.
Advanced LZS Technology (ALZS), Whitepaper, Hi/fn, Jun. 1, 1998, HIFN 002725-2727, 3 pgs.
Secure Tape Technology (STT) Whitepaper, Hi/fn, Jun. 1, 1998, HIFN 002728-2733, 6 pgs.
SSLRef 3.0 API Details, Netscape, Nov. 19, 1996, HIFN 002734-2778, 45 pgs.
LZS221-C Version 4 Data Compression Software Data Sheet , Hi/fn, 1994-97, HIFN 002779-2796, 18 pgs.
MPPC-C Version 4 Data Compression Software Data Sheet, Hi/fn, 1994-1997, HIFN 002797-2810, 14 pgs.
Magstar MP Hardware Reference B Series Models Document GA32-0365-01, 1996-1997, [IBM_1_601 pp. 1-338], 338 pages.
Magstar MP 3570 Tape Subsystem, Operator Guide, B-Series Models, 1998-1999, [IBM_1_601 pp. 339-525], 187 pages.
Preview, IBM Magstar 3590 Tape System Enhancements, Hardware Announcement, Feb. 16, 1999, [IBM_1_601 pp. 526-527], 2 pgs.
New IBM Magstar 3590 Models E11 and E1A Enhance Tape Drive Performance, Hardware Announcement, Apr. 20, 1999, [IBM_1_601 pp. 528-540] 13 pgs.
New IBM Magstar 3590 Model A60 Dramatically Enhances Tape Drive Performance, Hardware Announcement Jul. 27, 1999, [IBM_1_601 pp. 541-550] 10 pgs.
The IBM Magstar MP Tape Subsystem Provides Fast Access to Data, Sep. 3, 1996, Announcement No. 196-176, [IBM_1_601 pp. 551-563] 13 pgs.
IBM 3590 High Performance Tape Subsystem, Apr. 10, 1995, Announcement 195-106, [IBM_1_601 pp. 564-581] 18 pgs.
Standard ECMA-222 (Jun. 1995): ECMA—Standardizing Information and Communications Systems, Adaptive Lossless Data Compression Algorithm, [IBM_1_601 pp. 564-601] 38 pgs.
IBM 3590 and 3494 Revised Availability, Hardware Announcement Aug. 8, 1995, [IBM_743_1241 p. 1] 1 pg.
Direct Delivery of IBM 3494, 3466, and 3590 Storage Products, Hardware Announcement, Sep. 30, 1997, Announcement 197-297, [IBM_743_1241 pp. 2-3] 2 pgs.
IBM Magstar 3590 Enhances Open Systems, Hardware Announcement Feb. 9, 1996, Announcement 198-014, [IBM_743_1241 pp. 4-7] 4 pgs.
Hardware Withdrawal: IBM Magstar 3590 A00 Controller—Replacement Available, Announcement No. 197-267, Withdrawal Announcement, Dec. 9, 1997, [IBM_743_1241 p. 9] 1 pg.
IBM Magstar 3590 Tape Subsystem, Introduction and Planning Guide, Document No. GA32-0329007, [IBM_743_1241 pp. 10-499] 490 pgs.
NetMeeting 2.0 Reviewers Guide, Apr. 1997, [MSCS_298_339] 42 pgs.
Microsoft NetMeeting Compatible Products and Services Directory, Apr. 1997, [MSCS_242_297] 56 pgs.
Microsoft NetMeeting "Try This!" Guide, 1997, [MSCS_340_345] 6 pgs.
The Professional Companion to NetMeeting 2—The Technical Guide to Installing, Configuring, and Supporting NetMeeting 2.0 in Your Organization—Microsoft NetMeeting 2.0, 1996-97, [MSCS_2_241] 240 pgs.
CUSeeMe 3.1.2 User Guide, Nov. 1998, [RAD_1_220] 220 pgs.
MeetingPoint Conference Server Users Guide 3.0, Nov. 1997, [RAD_221_548] 328 pgs.
MeetingPoint Conference Server Users Guide 4.0.2, Dec. 1999, [RAD_549_818] 270 pgs.
MeetingPoint Conference Service Users Guide 3.5.1, Dec. 1998, [RAD_819_1062] 244 pgs.
Enhanced CUSeeMe—Authorized Guide, 1995-1996, [RAD_1063_1372] 310 pgs.
Meeting Point Reader File, Jun. 1999, [RAD_1437_1445] 9 pgs.
Press Release—White Pine Announces Launch of MeetingPoint Conferences Server, Oct. 9, 1997, [RAD_1738_1739] 2 pgs.
Press Release—Leading Network Service Providers Line Up to Support White Pine's MeetingPoint Conference Server Technology, Oct. 9, 1997, [RAD_1740_1743] 4 pgs.
Byte—A New MeetingPoint for Videoconferencing, Oct. 9, 1997, [RAD_1744_1750] 7 pgs.
Notice of Allowance in Commonly-Assigned U.S. Appl. No. 11/651,366, issued Apr. 10, 2009, 7 pgs.
Amendment under 37 C.F.R. §1.132 in Commonly-Assigned U.S. Appl. No. 11/651,366, filed Jul. 30, 2008, 18 pgs.
CCITT Draft Recommendation T.4, RFC 804, Jan. 1981, 12 pgs.
SNA Formats, IBM Corporation, 14^{th} Ed., Nov. 1993, 3 pgs.
Munteanu et al, "Wavelet-Based Lossless Compression Scheme with Progressive Transmission Capability," John Wiley & Sons, Inc., Int'l J. Imaging Sys. Tech., vol. 10, (1999) pp. 76-85.
Forchhammer and Jensen, "Data Compression of Scanned Halftone Images," IEEE Trans. Commun., vol. 42, Feb.-Apr. 1994, pp. 1881-1893.
Baker, K. et al., "Lossless Data Compression for Short Duration 3D Frames in Positron Emission Tomography," 0/7803-1487, May 1994, pp. 1831-1834.
Parties' Joint Claim Construction and Prehearing Statement Pursuant to P.R. 4-3, filed in *Realtime Data, LLC d/b/a/ IXO* v. *Packeteer, Inc. et al.,* Civil Action No. 6:08-cv-00144-LED; U.S. District Court for the Eastern District of Texas, (Feb. 18, 2009).
Declaration of Professor James A. Storer, Ph.D. relating to U.S. Patent No. 6,604,158, Mar. 18, 2009[d].
Declaration of Professor James A. Storer, Ph.D. relating to U.S. Patent No. 6,601,104, Mar. 18, 2009[b].
Declaration of Professor James A. Storer, Ph.D. relating to U.S. Patent No. 7,321,937, May 4, 2009.

Declaration of Professor James A. Storer, Ph.D. relating to U.S. Patent No. 6,624,761, May 4, 2009.
Declaration of Professor James A. Storer, Ph.D. relating to U.S. Patent No. 7,378,992, May 20, 2009.
Declaration of Professor James A. Storer, Ph.D. relating to U.S. Patent No. 7,161,506, May 26, 2009.
"Video Coding for Low Bit Rate Communication", International Telecommunication Union (ITU), Recommendation H.263, §3.4 (Mar. 1996) ("ITU H.263").
Rice, Robert F., "Some Practical Universal Noiseless Coding Techniques", Jet Propulsion Laboratory, Pasadena, California, JPL Publication 79-22, Mar. 15, 1979.
Anderson, J., et al. "Codec squeezes color teleconferencing through digital telephone lines", Electronics 1984, pp. 113-115.
Venbrux, Jack, "A VLSI Chip Set for High-Speed Lossless Data Compression", IEEE Trans. On Circuits and Systems for Video Technology, vol. 2, No. 44, Dec. 1992, pp. 381-391.
"Fast Dos Soft Boot", IBM Technical Disclosure Bulletin, Feb. 1994, vol. 37, Issue No. 2B, pp. 185-186.
"Operating System Platform Abstraction Method", IBM Technical Disclosure Bulletin, Feb. 1995, vol. 38, Issue No. 2, pp. 343-344.
Murashita, K., et al., "High-Speed Statistical Compression using Self-Organized Rules and Predetermined Code Tables", IEEE, 1996 Data Compression Conference.
Coene, W., et al. "A Fast Route for Application of Rate-distortion Optimal Quantization in an MPEG Video Encoder" Proceedings of the International Conference on Image Processing, US., New York, IEEE, Sep. 16, 1996, pp. 825-828.
Rice, Robert, "Lossless Coding Standards for Space Data Systems", IEEE 1058-6393/97, pp. 577-585.
Millman, Howard, "Image and video compression", Computerworld, vol. 33, Issue No. 3, Jan. 18, 1999, pp. 78.
"IBM boosts your memory", Geek.com [online], Jun. 26, 2000 [retrieved on Jul. 6, 2007], www.geek.com/ibm-boosts-your-memory/.
"IBM Research Breakthrough Doubles Computer Memory Capacity", IBM Press Release [online], Jun. 26, 2000 [retrieved on Jul. 6, 2007], www-03.ibm.com/press/us/en/pressrelease /1653.wss.
"ServerWorks to Deliver IBM's Memory eXpansion Technology in Next-Generation Core Logic for Servers", ServerWorks Press Release [online], Jun. 27, 2000 [retrieved on Jul. 14, 2000], http://www.serverworks.com/news/press/000627.html.
Abali, B., et al., "Memory Expansion Technology (MXT) Software support and performance", IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 287-301.
Franaszek, P. A., et al., "Algorithms and data structures for compressed-memory machines", IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 245-258.
Franaszek, P. A., et al., "On internal organization in compressed random-access memories", IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 259-270.
Smith, T.B., et al., "Memory Expansion Technology (MXT) Competitive impact", IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 303-309.
Tremaine, R. B., et al., "IBM Memory Expansion Technology (MXT)", IBM Journal of Research and Development, vol. 45, Issue No. 2, Mar. 2001, pp. 271-285.
Yeh, Pen-Shu, "The CCSDS Lossless Data Compression Recommendation for Space Applications", Chapter 16, Lossless Compression Handbook, Elsevier Science (USA), 2003, pp. 311-326.

* cited by examiner

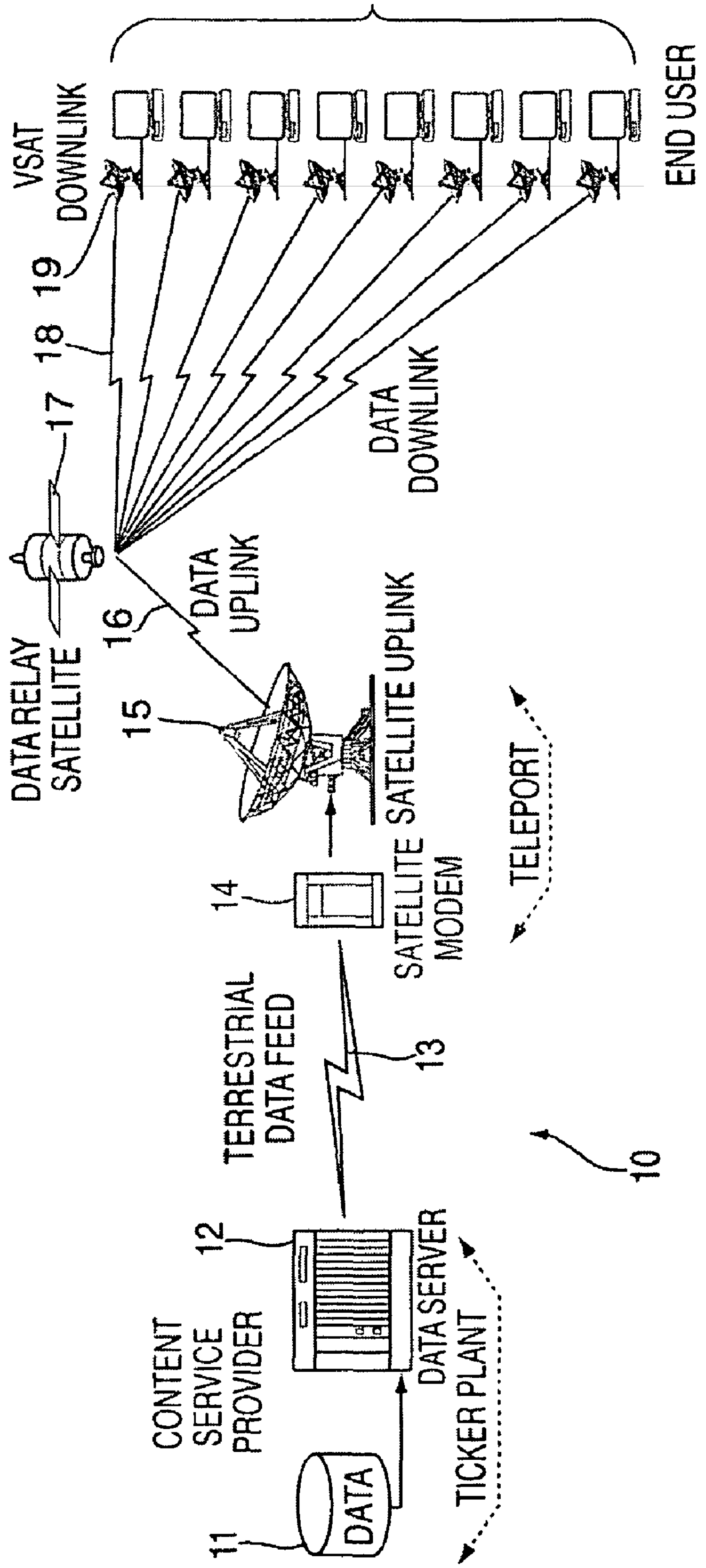

DATA
11
CONTENT SERVICE PROVIDER
12
DATA SERVER
TICKER PLANT
TERRESTRIAL DATA FEED
13
14
SATELLITE MODEM
TELEPORT
15
SATELLITE UPLINK
16
DATA UPLINK
DATA RELAY SATELLITE
17
DATA DOWNLINK
18
19
VSAT DOWNLINK
END USER
10

FIG. 1

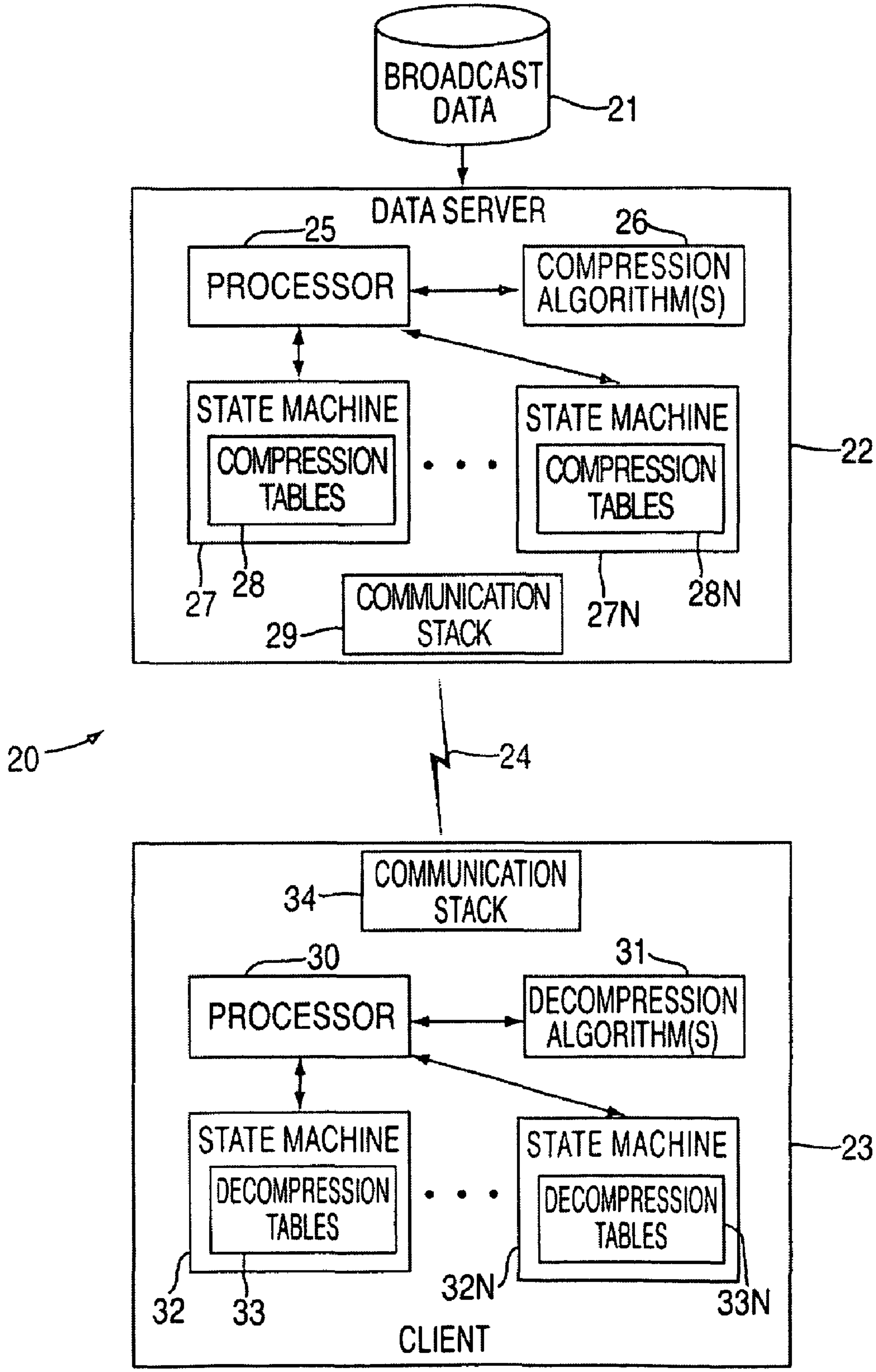

BROADCAST DATA
21
DATA SERVER
25
26
PROCESSOR
COMPRESSION ALGORITHM(S)
STATE MACHINE
COMPRESSION TABLES
STATE MACHINE
COMPRESSION TABLES
22
27
28
27N
28N
COMMUNICATION STACK
29
20
24
COMMUNICATION STACK
34
30
31
PROCESSOR
DECOMPRESSION ALGORITHM(S)
STATE MACHINE
DECOMPRESSION TABLES
STATE MACHINE
DECOMPRESSION TABLES
23
32
33
32N
33N
CLIENT

FIG. 2

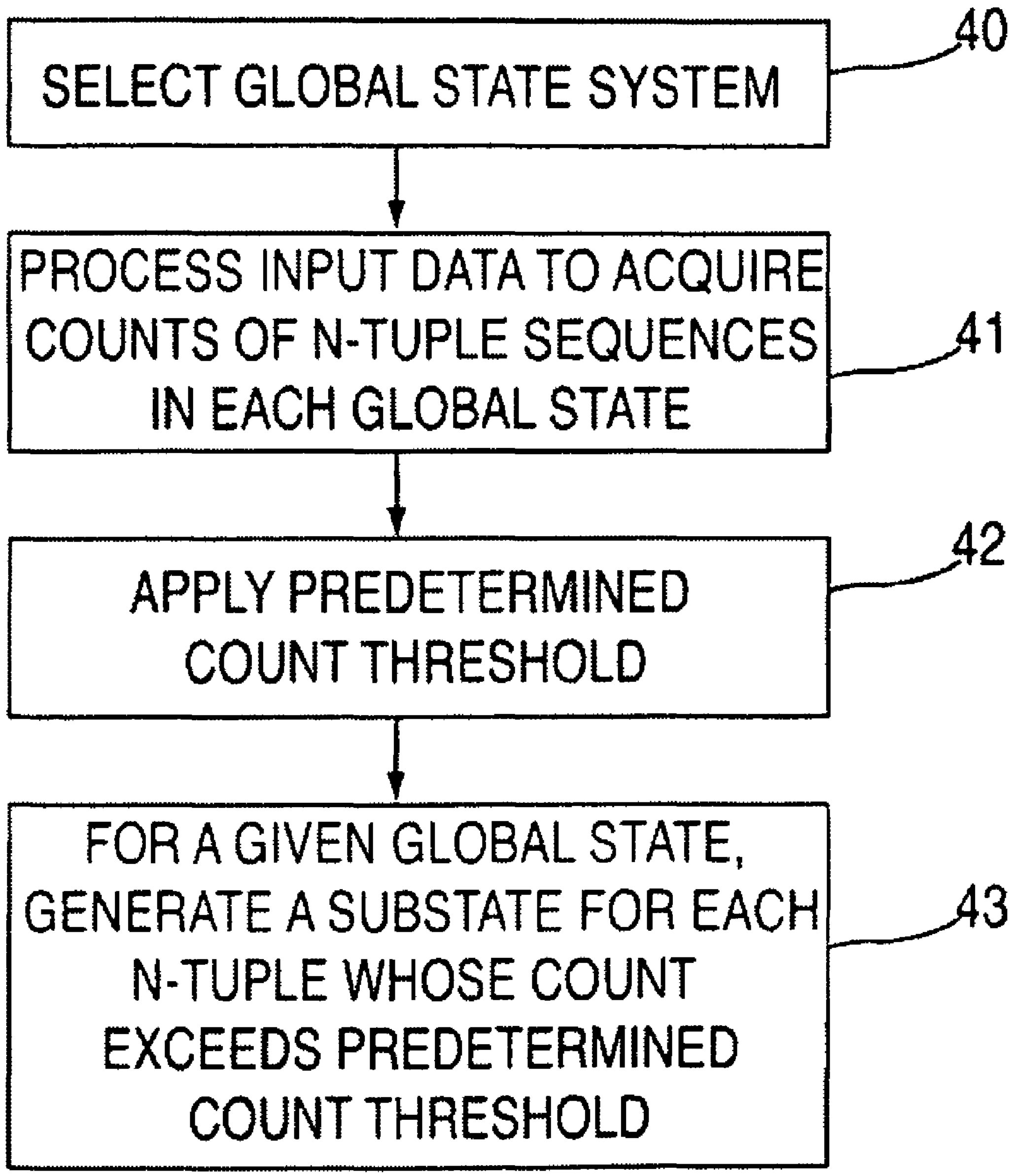

SELECT GLOBAL STATE SYSTEM
40
PROCESS INPUT DATA TO ACQUIRE COUNTS OF N-TUPLE SEQUENCES IN EACH GLOBAL STATE
41
APPLY PREDETERMINED COUNT THRESHOLD
42
FOR A GIVEN GLOBAL STATE, GENERATE A SUBSTATE FOR EACH N-TUPLE WHOSE COUNT EXCEEDS PREDETERMINED COUNT THRESHOLD
43

FIG. 3

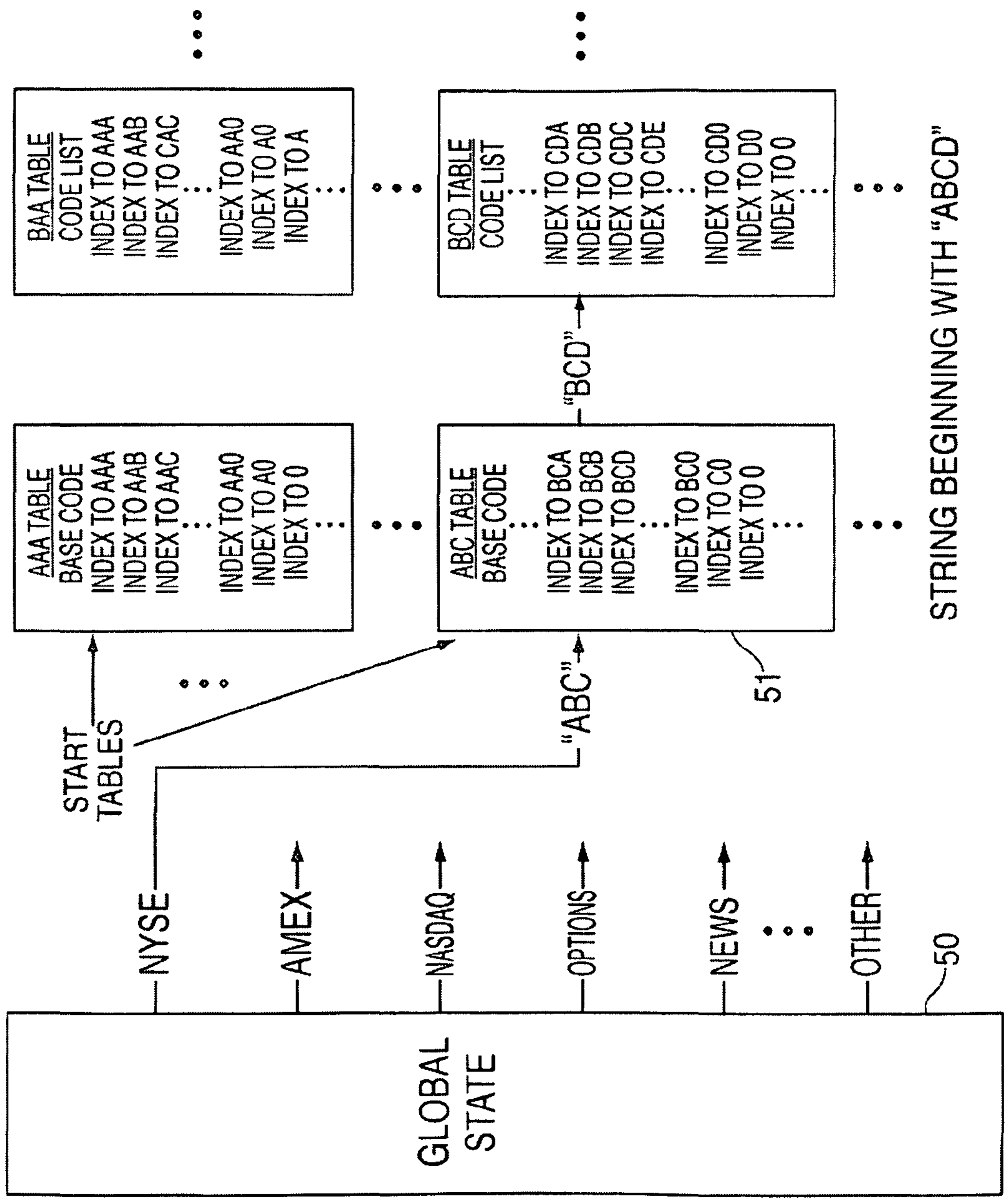
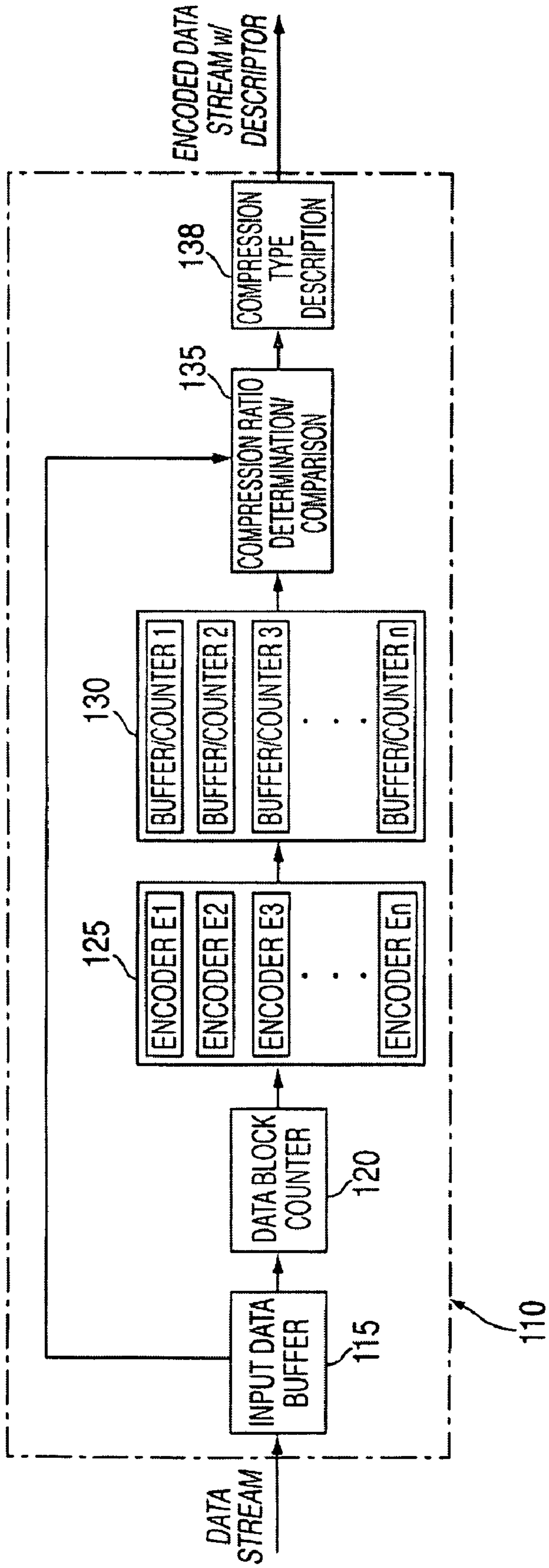

GLOBAL STATE
50
NYSE
AMEX
NASDAQ
OPTIONS
NEWS
OTHER
START TABLES
"ABC"
"BCD"
51
AAA TABLE
BASE CODE
INDEX TO AAA
INDEX TO AAB
INDEX TO AAC
INDEX TO AA0
INDEX TO A0
INDEX TO 0
BAA TABLE
CODE LIST
INDEX TO AAA
INDEX TO AAB
INDEX TO CAC
INDEX TO AA0
INDEX TO A0
INDEX TO A
ABC TABLE
BASE CODE
INDEX TO BCA
INDEX TO BCB
INDEX TO BCD
INDEX TO BC0
INDEX TO C0
INDEX TO 0
BCD TABLE
CODE LIST
INDEX TO CDA
INDEX TO CDB
INDEX TO CDC
INDEX TO CDE
INDEX TO CD0
INDEX TO D0
INDEX TO 0
STRING BEGINNING WITH "ABCD"

FIG. 4



DATA STREAM
INPUT DATA BUFFER
115
DATA BLOCK COUNTER
120
125
ENCODER E1
ENCODER E2
ENCODER E3
ENCODER En
130
BUFFER/COUNTER 1
BUFFER/COUNTER 2
BUFFER/COUNTER 3
BUFFER/COUNTER n
135
COMPRESSION RATIO DETERMINATION/ COMPARISON
138
COMPRESSION TYPE DESCRIPTION
ENCODED DATA STREAM w/ DESCRIPTOR
110

FIG. 5

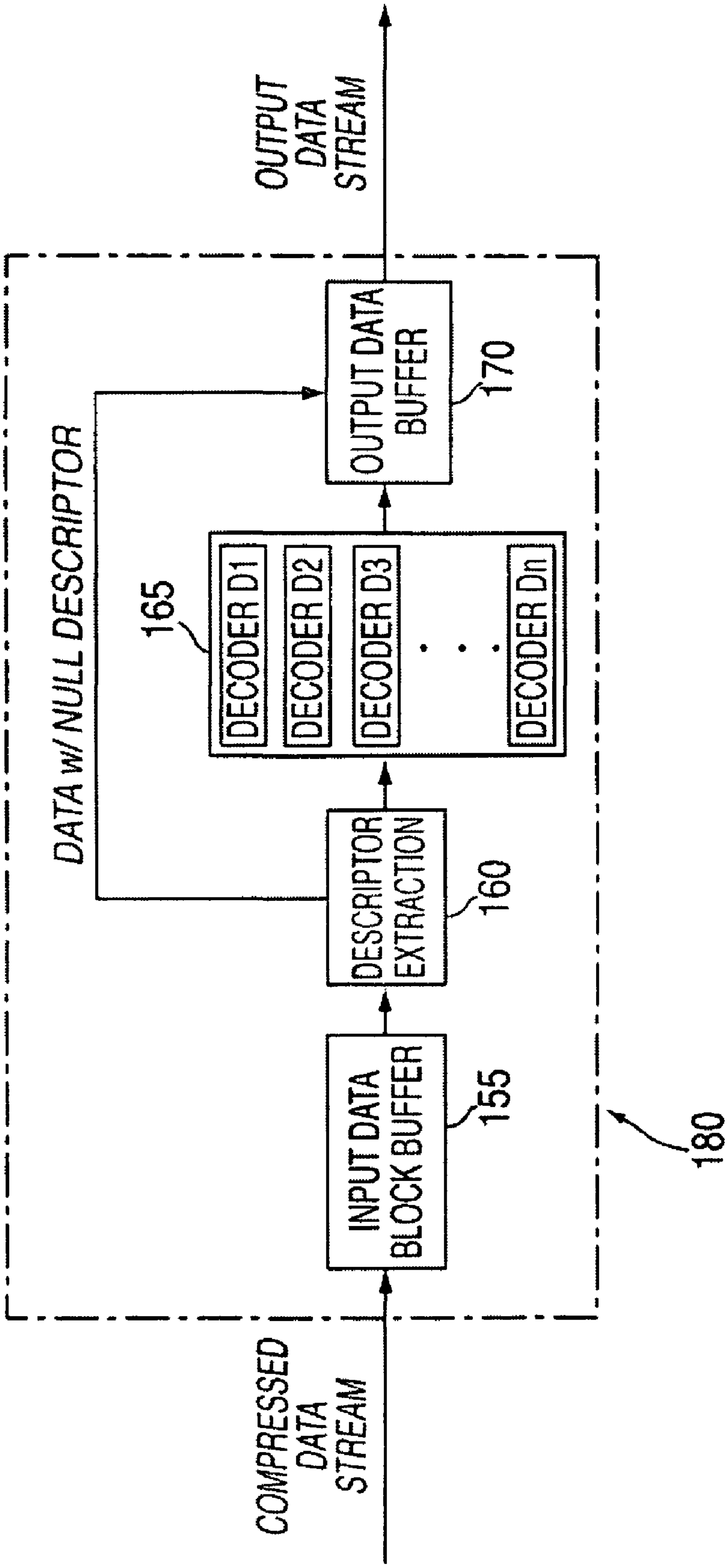

COMPRESSED DATA STREAM
INPUT DATA BLOCK BUFFER
155
DESCRIPTOR EXTRACTION
160
DATA w/ NULL DESCRIPTOR
165
DECODER D1
DECODER D2
DECODER D3
DECODER Dn
OUTPUT DATA BUFFER
170
OUTPUT DATA STREAM
180

FIG. 6

SYSTEM AND METHOD FOR DATA FEED ACCELERATION AND ENCRYPTION

CROSS REFERENCE TO RELATED APPLICATIONS

This is a continuation of patent application Ser. No. 10/434,305, filed on May 7, 2003, which is a Continuation-in-Part of U.S. patent application Ser. No. 09/969,987, filed on Oct. 3, 2001, which claims the benefit of U.S. Provisional Application No. 60/237,571, filed on Oct. 3, 2000, each of which are fully incorporated herein by reference. In addition, this claims the benefit of U.S. Provisional Application No. 60/378,517, filed on May 7, 2002, which is fully incorporated herein by reference.

TECHNICAL FIELD

The present invention relates generally to systems and method for providing data transmission, and in particular, to systems and method for providing accelerated transmission of data, such as financial trading data, financial services data, financial analytical data, company background data and news feeds, advertisements, and all other forms or information over a communication channel using data compression and decompression to provide data broadcast feeds, bi-directional data transfers, and all other forms of communication with or without security and effectively increase the bandwidth of the communication channel and/or reduce the latency of data transmission.

BACKGROUND

The financial markets and financial information services industry encompass a broad range of financial information ranging from basic stock quotations, bids, order, fulfillment, financial and quotations to analyst reports to detailed pricing of Treasury Bills and Callable bonds. Users of financial information can now generally be divided into three segments—Traders, Information Users and Analytics Users, although some users constitute components from one or more of these categories.

Traders utilize data from financial markets such as NASDAQ, the American Stock Exchange, the New York Stock Exchange, the Tokyo Exchange, the London Exchange, the Chicago Options Board, and similar institutions that offer the ability to buy and sell stocks, options, futures, bonds, derivatives, and other financial instruments. The need for vast quantities of information is vital for making informed decisions and executing optimal transactions

Thus given the importance of receiving this information over computer networks, an improved system and method for providing secure point-to-point solution for transparent multiplication of bandwidth over conventional communication channels is highly desirable.

For example, with the introduction of Nasdaq's next generation trading system SuperMontage, Nasdaq will offer market data users an unparalleled view into the activity, liquidity, and transparency of the Nasdaq market.

For example, currently Nasdaq provides each market participant's best-attributed quotation in each stock in which it makes a market. This system known as SuperMontage allows Nasdaq to accept multiple orders from each market participate in each stock for execution within SuperMontage. Nasdaq offers that data, with multiple levels of interest from individual market participants, through new data services.

Nasdaq provides this data on both an aggregated and a detailed basis for the top five price levels in SuperMontage. This data is currently offered through market data vendors and broker/dealer distributors via the following four entitlement packages:

QuoteView[SM] Each SuperMontage participant's best bid and offer, as well as the best bid and offer available on SuperMontage.

DepthView[SM] The aggregate size, by price level, of all Nasdaq market participants' attributed and unattributed quotations/orders that are in the top five price levels in SuperMontage.

PowerView[SM] Bundled QuoteView and DepthView.

TotalView[SM] PowerView plus all Nasdaq market participants' attributed quotations/orders that are in the top five price levels in SuperMontage, in addition to the aggregate size of all unattributed quotes/orders at each of the top five price levels.

The NASDAQ SuperMontage trading system has been cited to be representative of trend for explosive growth in the quantity of information for all emergent and future trading and financial information distribution systems. Increases in processing power at the end user sites will allow traders, analysts, and all other interested parties to process substantially larger quantities of data in far shorter periods of time, increasing the demand substantially.

The ever increasing need for liquidity in the financials markets, coupled with the competitive pressures on reducing bid/ask spreads and instantaneous order matching/fulfillment, along the need for synchronized low latency data dissemination makes the need for the present invention ever more important. Depth of market information, required to achieve many of these goals requires orders of magnitude increases in Realtime trade information and bid/ask pricing (Best, 2^{nd} best, . . .).

A fundamental problem within the current art is the high cost of implementing, disseminating, and operating trading systems such as SuperMontage within the financial services industry. This is in large part due to the high bandwidth required to transfer the large quantities of data inherent in the operation of these systems. In addition the processing power required to store, transmit, route, and display the information further compounds cost and complexity.

This fundamental problem is in large part the result of utilizing multiple simultaneous T1 lines to transmit data. The data must be multiplexed into separate data streams, transmitted on separate data lines, and de-multiplexed and checked. Software solutions have high latency and cost while hardware solutions have even higher cost and complexity with somewhat lower latency. In addition the synchronization and data integrity checking require substantial cost, complexity, inherent unreliability, and latency. These and other limitations are solved by the present invention.

Further compounding this issue is a globalization and consolidation taking place amongst the various financial exchanges. The emergence of localized exchanges (ECNS—Electronic Computer Networks) coupled with the goal of 24 hour/7 day global trading will, in and of itself, drive another exponential increase in long haul international bandwidth requirements, while ECNs and other localized trading networks will similarly drive domestic bandwidth requirements. Clearly long haul links are orders of magnitude more expensive than domestic links and the value and significance of the present invention is at least proportionately more important.

Information users range from non-finance business professionals to curious stock market investors and tend to seek basic financial information and data. Analytical users on the

other hand, tend to be finance professionals who require more arcane financial information and utilize sophisticated analytical tools to manipulate and analyze data (e.g. for writing option contracts).

Historically, proprietary systems, such as Thomson, Bloomberg, Reuters and Bridge Information, have been the primary electronic source for financial information to both the informational and analytical users. These closed systems required dedicated telecommunications lines and often product-specific hardware and software. The most typical installations are land-based networking solutions such as T1, or ISDN, and satellite-based "wireless" solutions at speeds of 384 kbps.

Latency of financial data is critical to the execution of financial transactions. Indeed the more timely receipt of financial data from various sources including the New York Stock Exchange, American Stock Exchange, National Association of Securities Dealers (NASDAQ), Options Exchange, Commodities Exchanges, and Futures presents a fundamental advantage to those who trade. Latency is induced by the long time taken transmit and receive uncompressed data or to compress and encrypt data prior to transmission, along with the associated time to decrypt and decompress. Often current methods of encryption and compression take as much or substantially more time than the actual time to transmit the uncompressed, unencrypted data. Thus another problem within the current art is the latency induced by the act of encryption, compression, decryption, and decompression. The present invention overcomes this limitation within the current art.

Modern data compression algorithms suffer from poor compression, high latency, or both. Within the present art algorithms such as Lempel-Ziv, modified/embellished Lempel-Ziv, Binary Arithmetic, and Huffman coding are essentially generic algorithm having a varied effectiveness on different data types. Also small increases in compression to the negentropy limit of the data generally require exponentially greater periods of time and substantially higher latency. Negentropy is herein defined as the information content within a given piece of data. Generic algorithms are currently utilized as data types and content format is constantly changed within the financial industry. Many changes are gradual however there are also abrupt changes, such as the recent switch to decimalization to reduce granularity that has imposed substantial requirements on data transmission bandwidth infrastructure within the financial industry. Thus another problem within the current art is the high latency and poor compression due to the use of generic data compression algorithms on financial data and news feeds. This limitation is also overcome by the present invention.

Within the financial and news feeds, data is often segregated into packets for transmission. Further, in inquiry-response type systems, as found in many financial research systems, the size of request packets and also response packets is quite small. As such, response servers often wait for long periods of time (for example 500 msec) to aggregate data packets prior to transmission back to the inquirer. By aggregating the data, and then applying compression, somewhat higher compression ratios are often achieved. This then translates to lower data communications costs or more customers served for a given amount of available communications bandwidth. Thus another problem within the current art is the substantial latency caused by aggregating data packets due to poor data compression efficiency and packet overhead. This limitation is also solved by the present invention.

Another problem within the current art is the need for data redundancy. Currently many trading systems utilize two independent links to compare data to verify integrity. Second, the bandwidth of discrete last mile links, typically T1s, is limited to 1.5 Megabits/second.

Increases in bandwidth beyond this point require complex protocols to fuse data from multiple links, adding cost and complexity, while also increasing latency and inherent data error rates. This limitation is also solved by the present invention.

Another limitation within the current art is that nearly all financial institutions use one or more T1 lines to transfer information to and from their customers. While the costs of bandwidth have moderately decreased over recent years this trend is slowing and the need forever increased bandwidth will substantively overshadow any future reductions. Indeed with the recent fall-out of the telecommunications companies the data communications price wars will end and we could easily see an increase in the cost of bandwidth. US Domestic T1 lines currently range from several hundred dollars to upwards of a thousand dollars per link, dependent upon quantity of T1 lines purchased, geographic location, length of connection, and quality/conditioning of line. Fractional T1 lines may also be purchased in 64 Kilobit/second increments with some cost savings.

A standard T1 line transmits data at a rate of 1.544 megabits per second. Accounting for framing and data transmission overhead this means that a T1 line is capable of transmitting a 150 Kilobytes per second. While 30× faster than a modem line (which provides only 5 kilobytes per second), both are relatively slow in relation to any reasonable level of information flow. For example, transferring the contents of data on a single CDROM would take well over an hour!

Thus it is likely that the capacity of many existing T1 lines will be exceeded in the near future. For our current example let's assume that we need to double the capacity of a T1 line. Normally this is done by adding a second T1 line and combining the contents of both with Multi-Link Point to Point Protocol (MLPP) or another relatively complex protocol. Within the current art this is neither necessary nor desirable. In fact any increase over the current limitation of a T1 line results in the addition of a second line. This limitation is overcome by the present invention.

Another limitation with the current art is the extraordinary bandwidth required for real-time (hot) co-location processing which has been dramatically increased as a result of the acts of terror committed against the United States of America on Sep. 11, 2001. In order for the redundancy of any co-location to be effective, it must be resident in a geographically disparate location; this could be a different state, a different coast, or even a different country. The trend towards globalization will further compound the need for the ability to simultaneously process transactions at geographically diverse co-locations.

It is a widely known fact within the financial industry that the overall throughput of transactions is governed by the bandwidth and latency of the co-location data link, along with delays associated with synchronization, i.e. the transaction must be complete at both locations and each location must know that the other location is complete before the transaction is finalized.

High bandwidth links such as T3's are often utilized as part of this backbone structure. A single T3 line has the bandwidth of Twenty-Eight T1 lines (28×1.544=43.232 megabits/second). Thus, in the best case, a T3 line is capable of transmitting 5.4 megabytes/second. By way of comparison, the contents of a single CDROM may be transferred in approximately two minutes with a T3 link. As stated earlier, a single T1 line would take over an hour to transmit the same quantity of data.

The volume of real-time data that is required to operate any major financial institution is staggering by comparison. To deal with this issue only critical account and transaction information is currently processed by co-locations in real-time. In fact, many institutions use batch mode processing where the transactions are only repeated "backed up" at the co-locations some time period later, up to 15 minutes or longer. The limitation of highly significant bandwidth and/or long delays with co-location processing and long latency times is solved by the present invention.

Thus given the importance of receiving financial information over computer networks, an improved system and method for providing secure point-to-point solution for transparent multiplication of bandwidth over conventional communication channels is highly desirable.

As previously stated, these and other limitations within the current art are solved by the present invention.

SUMMARY OF THE INVENTION

The present invention is directed to systems and methods for providing accelerated data transmission, and in particular to systems and methods of providing accelerated transmission of data, such as financial trading data, financial services data, financial analytical data, company background data, news, advertisements, and all other forms of information over a communications channel utilizing data compression and decompression to provide data transfer (secure or non-secure) and effectively increase the bandwidth of the communication channel and/or reduce the latency of data transmission. The present invention is universally applicable to all forms of data communication including broadcast type systems and bi-directional systems of any manner and any number of users or sites.

These and other aspects, features and advantages, of the present invention will become apparent from the following detailed description of preferred embodiments that is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF PREFERRED EMBODIMENTS

FIG. **1** is a block diagram of a system in which the present invention may be implemented for transmitting broadcast data;

FIG. **2** is a block diagram of a system and method for providing accelerated transmission of data over a communication channel according to an embodiment of the present invention;

FIG. **3** is a flow diagram illustrating a method for generating compression/decompression state machines according to one aspect of the present invention;

FIG. **4** is a diagram illustrating an exemplary encoding table structure according to the present invention, which may be generated using the process of FIG. **3**.

FIG. **5** is a diagram of a system/method for providing content independent data compression, which may be implemented for providing accelerated data transmission according to the present invention; and

FIG. **6** is a diagram of a system/method for providing content independent data decompression, which may be implemented for providing accelerated data transmission according to the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention is directed to systems and methods for providing accelerated transmission of broadcast data, such as financial data and news feeds, over a communication channel using data compression and decompression to provide secure transmission and transparent multiplication of communication bandwidth, as well as reduce the latency associated with data transmission of conventional systems.

In this disclosure, the following patents and patent applications, all of which are commonly owned, are fully incorporated herein by reference: U.S. Pat. Nos. 6,195,024, issued on Feb. 27, 2001, and 6,309,424, issued on Oct. 30, 2001 and U.S. patent application Ser. Nos. 10/076,013 filed on Feb. 13, 2002, 10/016,355, filed on Oct. 29, 2001, 09/481,243 filed on Jan. 11, 2000, and 09/266,394 filed on Mar. 11, 1999.

In general, the term "accelerated" data transmission refers to a process of receiving a data stream for transmission over a communication channel, compressing the broadcast data stream in real-time (wherein the term "real time" as used herein collectively refers to substantially real time, or at real time, or greater than real time) at a compression rate that increases the effective bandwidth of the communication channel, and transmitting the compressed broadcast data over the communication channel. The effective increase in bandwidth and reduction of latency of the communication channel is achieved by virtue of the fast than real-time, real-time, near real time, compression of a received data stream prior to transmission.

For instance, assume that the communication channel has a bandwidth of "B" megabytes per second. If a data transmission controller is capable of compressing (in substantially real time, real time, or faster than real time) an input data stream with an average compression rate of 3:1, then data can be transmitted over the communication channel at an effective rate of up to 3*B megabytes per second, thereby effectively increasing the bandwidth of the communication channel by a factor of three.

Further, when the receiver is capable decompressing (in substantially real time, real time, or faster than real time) the compressed data stream at a rate approximately equal to the compression rate, the point-to-point transmission rate between the transmitter and receiver is transparently increased. Advantageously, accelerated data transmission can mitigate the traditional bottleneck associated with, e.g., local and network data transmission.

If the compression and decompression are accomplished in real-time or faster, the compressed, transmitted and decompressed data is available before the receipt of an equivalent uncompressed stream. The "acceleration" of data transmission over the communication channel is achieved when the total time for compression, transmission, and decompression, is less than the total time for transmitting the data in uncompressed form. The fundamental operating principle of data acceleration is governed by the following relationship:

$$[T_{Compress} + T_{Transmit\ Accelerated} + T_{Decompress}] < T_{Transmit\ w/o\ Compression} \quad \text{EQ [1]}$$

Where:

$T_{Compress}$ = Time to Compress a Packet of Data

$T_{Transmit\ Accelerated}$ = Time to Transmit Compressed Data Packet

$T_{Decompress}$ = Time to Decompress the Compressed Data Packet

$T_{Transmit\ w/o\ Compression}$ = Time to Transmit the Uncompressed (Original) Data Packet

As stated in Equation [1] above, if the time to compress, transmit, and decompress a data packet is less than the time to transmit the data in original format, then the delivery of the data is said to be accelerated.

In the above relationship, a fundamental premise is that all information is preferably fully preserved. As such, lossless data compression is preferably applied. While this disclosure is directed to transmission of data in financial networks, for example, the concept of "acceleration" may be applied to the storage and retrieval of data to any memory or storage device using the compression methods disclosed in the above-incorporated U.S. Pat. Nos. 6,195,024 and 6,309,424, and U.S. application Ser. No. 10/016,355, and the storage acceleration techniques disclosed in the above-incorporated application Ser. No. 09/481,243 and 09/266,394.

Returning to Equation [1], data acceleration depends on several factors including the creation of compression and decompression algorithms that are both effective (achieve good compression ratios) and efficient (operate rapidly with a minimum of computing processor and memory resources).

Rearranging the terms of Equation [1] we can see that the total time to transmit data in an "accelerated" form (transmit compressed data (is the sum of the original time to transmit the data in an uncompressed fashion divided by the actual compression ratio achieved, plus the time to compress and decompress the data.

$$T_{Transmit\ Accelerated}=[T_{ransmit\ w/o\ Compression}/\mathrm{CR}]+T_{Compress}+T_{Decompress} \quad \text{EQ [2]}$$

Where:

CR=Compression Ratio

Thus the latency reduction is the simple arithmetic difference between the time to transmit the original data minus the total time to transmit the accelerated data (per Equation 2 above), resulting in:

$$T_{Latency\ Reduction}=T_{Transmit\ w/o\ Compression}-T_{Transmit\ Accelerated} \quad \text{EQ [3]}$$

And finally the achieved "Acceleration Ratio" is defined as:

$$\text{Acceleration Ratio}=T_{Transmit\ w/o\ Compression}/T_{Transmit\ Accelerated} \quad \text{EQ [4]}$$

A number of interesting observations come to light from these relatively simple algebraic relationships and are implemented within the present invention:

Compression Ratio The present inventions achieve a consistent reduction in latency. The data compression ratio is substantial and repeatable on each data packet.

Compression Rate The present invention achieves a consistent reduction in latency. Both the time to compress and decompress the data packet must be an absolute minimum, repeatable on each data packet, and always within predefined allowable bounds.

Packet Independence: The present invention has no packet-to-packet data dependency. By way of example, in UDP and Multicast operations there are no guarantees on delivery of data packets, nor on the order of delivered data packets. IP data packets, similarly, have no guarantee on the order of delivery also. Thus algorithms that rely on dictionaries (Zlib, Glib, Lempel Ziv, etc.) are inherently unreliable in any financial real-world financial data applications.

It is to be understood that the present invention may be implemented in various forms of hardware, software, firmware, or a combination thereof. Preferably, the present invention is implemented on a computer platform including hardware such as one or more central processing units (CPU) or digital signal processors (DSP), a random access memory (RAM), and input/output (I/O) interface(s). The computer platform may also include an operating system, microinstruction code, and dedicated processing hardware utilizing combinatorial logic or finite state machines. The various processes and functions described herein may be either part of the hardware, microinstruction code or application programs that are executed via the operating system, or any combination thereof.

It is to be further understood that, because some of the constituent system components described herein are preferably implemented as software modules, the actual system connections shown in the Figures may differ depending upon the manner in that the systems are programmed. General purpose computers, servers, workstations, personal digital assistants, special purpose microprocessors, dedicated hardware, or and combination thereof may be employed to implement the present invention. Given the teachings herein, one of ordinary skill in the related art will be able to contemplate these and similar implementations or configurations of the present invention.

It should be noted that the techniques, methods, and algorithms and teachings of the present invention are representative and the present invention may be applied to any financial network, trading system, data feed or other information system.

FIG. **1** is a diagram illustrating a system in which the present invention may be implemented. The system **10** comprises content **11** and data server **12** associated with a service provider of broadcast data. The content **11** comprises information that is processed by the data server **12** to generate a broadcast, e.g., a news feed or financial data feed. As explained in further detail below, the data server **12** employs data compression to encode/encrypt the broadcast data **11** prior to transmission over various communication channels to one or more client site systems **20** of subscribing users, which comprise the necessary software and hardware to decode/decrypt the compressed broadcast data in real-time. In the exemplary embodiment of FIG. **1**, the communication channels comprise a landline **13** that feeds the compressed broadcast data to a satellite system comprising modem **14** and an uplink system **15**, which provides a data uplink **16** to a relay **17**. The relay **17** provides data downlinks **18** to one or more downlink systems **19**.

Advantageously, the proprietary software used by the data server **12** to compress the data stream in real-time and software used by the workstations **19** to decompress the data stream in real-time effectively provides a seamless and trans-

parent increase in the transmission bandwidth of the various communication channels used, without requiring modification of existing network infrastructure.

Referring now to FIG. **2**, a block diagram illustrates a system/method for providing accelerated transmission of data according to one embodiment of the present invention. More specifically, FIG. **2** illustrates embodiments of a broadcast data server (transmitter) and client system (receiver) for implementing accelerated transmission and real-time processing of broadcast data. Broadcast data **21** (comprising one or more different broadcast types) is processed by data server **22** prior to transmission to client **23** over a communication channel **24**. The data server **22** utilizes a processor **25** (e.g., microprocessor, digital signal processor, etc.) for executing one or more compression algorithms **26** for compressing (in real-time) the broadcast data **21** prior to transmission. In preferred embodiments, compression is achieved using Huffman or Arithmetic encoding, wherein one or more state machines **27**-**27***n* are constructed based on a-priori knowledge of the structure and content of one or more given broadcast and data feeds.

As explained in further detail below, each state machine **27**-**27***n* comprises a set of compression tables that comprise information for encoding the next character (text, integer, etc.) or sequence of characters in the broadcast data feed, as well as pointers which point to the next state (encoding table) based on the character or character sequence. As explained in greater detail below, a skeleton for each state machine **27**-**27***n* (nodes and pointers) is preferably built by finding sequences of characters (n-tuples) that frequently appear in a given data input. Once a skeleton has been determined, a large set of data is processed through the system and counts are kept of character n-tuples for each state. These counts are then used to construct the compression tables associated with the state machine to provide statistical compression. The compressed data is transmitted over the communication channel **24** via a communication stack using any suitable protocol (e.g., RTP (real time protocol) using RTCP (real-time control protocol), TCP/IP, UDP, or any real-time streaming protocol with suitable control mechanism).

Similarly, the client **23** comprises a processor **30** for executing one or more decompression algorithms **31**. Depending one the data feed type, one of a plurality of decompression state machines **32**-**32***n* are used to decompress the compressed data stream received by the client **23** via communication stack **34**. Each state machine **32**-**32***n* comprises a set of decompression tables **33**-**33***n* that comprise information for decode the next encoded character (or symbol) or sequence of symbols in the compressed broadcast data feed, as well as pointers which point to the next state based on the symbol or symbol sequence. For each compression state machine **27**-**27***n* in the data server, a corresponding decompression state machine **32**-**32***n* is needed in the client **23** to decompress the associated data stream.

Advantageously, a compression/decompression scheme according to the present invention using Huffman or Arithmetic encoding provides secure transmission via de facto or virtual "encryption" in a real-time environment. Indeed, virtual encryption is achieved by virtue of the fast, yet complex, data compression using Huffman tree, for example, without necessarily requiring actual encryption of the compressed data and decryption of the compressed data. Because of the time-sensitive nature of the market data, and the ever-changing and data-dependent nature of the arithmetic scheme, decryption is virtually impractical, or so complex and useless as to render the data worthless upon eventual decoding.

However, data compression using Huffman or Arithmetic encoding yields encoded data that is very difficult to decode than current encryption schemes such as plain text or simple bit shuffling codes as currently used by broadcast service providers. An attacker must have the compression model and the tables used to compress the data stream to be able to obtain useful information from it. Thus, at one level of security, the client-side decompression tables are preferably stored in encrypted form and are decrypted on being loaded into the processor **30** (e.g., general purpose processor, DSP, etc.) using an encryption/decryption key that is validated for a subscribing user. In this manner, a client will be unable to use the tables on other processors or sites or after terminating a service contract.

Since Huffman compression uses the same bit code for a character each time it appears in a given context, an attacker with a very large data set of compressed and uncompressed data could possibly reconstruct the tables, assuming the overall model were known. Arithmetic compression, on the other hand, generates different bit patterns for the same character in the same context depending on surrounding characters. Arithmetic encoding provides at least an order of magnitude more difficult to recover the tables from the compressed and uncompressed data streams.

The following is a detailed discussion of a compression scheme using Huffman or Arithmetic encoding for providing accelerated transmission of broadcast data according to one aspect of the present invention. It is to be appreciated that the present invention is applicable with any data stream whose statistical regularity may be captured and represented in a state machine model. For example, the present invention applies to packetized data streams, in which the packets are limited in type format and content.

In one embodiment using Huffman or Arithmetic encoding, each character or character sequence is encoded (converted to a binary code) based on the frequency of character or character sequence in a given "context". For a given context, frequently appearing characters are encoded with few bits while infrequently appearing characters are encoded with more bits. High compression ratios are obtained if the frequency distribution of characters in most contexts is highly skewed with few frequently appearing characters and many characters seldomly (or never) appear.

Referring now to FIG. **3**, a flow diagram illustrates a method for generating compression/decompression state machines according to one aspect of the present invention. The "context" in which a character (or character sequence) is encoded in a given broadcast stream is based on a "global state" that represents packet type and large-scale structure and the previous few characters. The first step in building a compression scheme involves selecting a global state system based on the packet structure of the broadcast model (step **40**). More specifically, a global state system is constructed based on a priori knowledge of the data stream model, e.g., the packet type frequency and structure of the broadcast model. By way of example, one model for financial data may comprise four global states representing: a beginning of packet, an options packet, a NYSE (New York Stock Exchange) packet and some other packet type. Further, additional codes may be added to the encoding tables to indicate global state transitions (e.g., for an end of packet code in the broadcast model). If there is internal structure to packets, such as a header with different statistics than the body, additional global states could be added.

Once a global state system is selected, training samples from an associated data stream are passed through the global model to acquire counts of frequencies of the occurrence of

n-tuple character sequences ending in each of the model states (step **41**). In a preferred embodiment, the n-tuples comprise character sequences having 1, 2 and 3 characters. Using the acquired counts, sub-states (or "local states") of the predefined global states are constructed based on previous characters in the data stream. A local state may depend on either none, 1, 2, or 3 (or more) previous characters in the stream. To provide a practical limitation, a predetermined count threshold is preferably applied to the count data (step **42**) and only those sequences that occur more often than the count threshold are added as local states (step **43**). For example, if a three-character sequence does not occur sufficiently frequently, the count for the last two characters is tested, etc.

It is to be understood that any character sequence length "n" may be implemented depending on the application. The longer the allowed character sequence, the more memory is needed to store the encoding tables and/or the lower the count threshold should be set.

As samples of the data are passed through the state model, character (and transition code) counts for each context are accumulated. These counts are used to build the Huffman or Arithmetic coding tables. The construction of the global and local models is an iterative process. The count threshold for forming local states can be adjusted depending on the application. For instance, a larger threshold will result in less local states but less compression as well. Further, a comparison of statistics in local or global states may suggest adding or deleting global states.

The construction of the global model requires knowledge of the data stream packet structure. The construction of the local states is automatic (once the threshold is set).

FIG. **4** is a diagram of an exemplary state diagram (or encoding table structure) according to the present invention, which may be generated using the process of FIG. **3**.

As noted above with reference to FIGS. **1** and **2**, a compression scheme according to the present invention may be implemented in any system to provide accelerated data transmission to multiple client site systems. Preferably, the client site systems may connect at any time, so minimal immediate history may be used (since a newly connected site must be able to pick up quickly). A system according to an embodiment of the present invention uses statistical compression (Huffman or Arithmetic coding) using fixed (or adaptive) tables based on the statistics of a data feed sample. As noted above, it has been determined that the statistical compression schemes described herein are well adapted for use with structured data streams having repetitive data content (e.g., stock symbols and quotes, etc.) to provide fast and efficient data compression/decompression.

The following discussion provides further details regarding the preparation of statistical-based encoding tables and their use for compression/decompression according to the present invention. During a data compression process, the selection of which encoding table to use for compression is preferably based on up to n (where n is preferably equal to 3) preceding characters of the message. In an exemplary broadcast model tested by the present inventors, a data stream comprises messages that begin with an ID code in the range 0-31 with the remainder of the message being characters in the range 32-127. It was found that approximately half of the messages in a given sample began with ID code 0x0c and half of the remainder began with ID code 0x0f. Thus, a separate encoding table is preferably used for a message ID code. Further, separate table sets are used for messages beginning with 0x0c and with 0x0f, with the remaining messages lumped together in another table.

Each table has an additional termination code. The termination code in a "start table" indicates the end of a compression block. The termination code in all other tables indicates the end of the message. Thus, the start table comprises 33 entries and all other tables have 97 entries.

Using one table for each 3-character context would require prohibitive amounts of memory. For example, a complete one-character context would require 33+3*97=324 tables. Then, a complete two-character context would require 324*97=31,428 tables. And finally, a complete three-character context would require 324*97*97=3,048,516 tables. Preferably, as described above, the application of a count threshold at each context size reduces the amount of tables. Only when a context occurs at greater than the threshold rate in the sample will a table be created for that context.

Each table entry includes a link to the next table to be used. For instance, in an "abc" context table, the entry for next character "d" would point to the "bcd" table, if such table was created. If such table was not created, the entry for next character "d" would point to the "cd" table, if such table existed. If no "cd" table exists, the "d" table would be used and if that fails, a base table for the message type would be used.

For a client site system to pick up the broadcast feed at any time, clearly identifiable synchronization points are preferably included in the compressed data stream. In a preferred embodiment, data is compressed in blocks with each block comprising some number of complete messages. Preferably, each compressed block ends with at least four bytes with each bit being logic 1 and no interior point in the compressed block will comprise 32 consecutive 1 bits. The compressed block preferably begins with two bytes giving the decompressed size of the block shifted to guarantee that the first byte of the compressed block is not all 1's. Thus, to achieve synchronization, the client site system can scan the input compressed data stream for 4 bytes of 0xff, wherein the next byte not equal to 0xff is deemed the start of a compressed block. In other words, the receiver will accumulate the compressed data until at least a sequence of 4 bytes each having a value of 0xff is detected in the input stream, at which point decompression will commence on the compressed input stream.

In another embodiment of the present invention, if a compressed block is more than 6 bytes longer than the uncompressed data, the data block is transmitted uncompressed preceded by the shifted two-byte count with the high bit set and trailed by 4 bytes of 0xff.

The following is discussion of a method for preparing Huffman Tables according to one aspect of the present invention. The Huffman codes generated by a conventional optimal algorithm have been modified in various ways in accordance with the present invention. First, in order that there not be 32 consecutive one bits in the data stream except at the end of a compression block, a termination code in each table comprises all 1 bits.

Further, to reduce space required for decompression tables, and ensure no sequence of 32 1 bits, each code is preferably decoded as follows:

a) The first 7 bits are used to index into a table. If the character code is no more than 7 bits, it can be read directly;

b) otherwise, some number N of initial bits is discarded and the next 7 bits are used to index a second table to find the character.

Based on these steps, preferably, no character code can use more than 14 bits and all codes of more than 7 bits must fit into the code space of the N initial bits. If N is 3, for instance, then no code can use more than 10 bits.

To achieve this, the code space required for all optimal codes of more than 7 bits is first determined, following by a determining the initial offset N. Every code comprising more than N+7 bits is preferably shortened, and other codes are lengthened to balance the code tree. It is possible that this may cause the code space for codes over 7 bits to increase so that N may need to be decreased. Preferably, this process is performed in a manner that causes minimal reduction in the efficiency of the codes.

The above modifications to convention optimal algorithm yields codes in which no non-termination code ends in more than 7 1 bits, no non-termination code begins with more than 6 1 bits, no termination code is more than 14 1 bits and no non-termination packet start code begins with more than 5 1 bits. Thus, in the middle of a packet, a sequence of no more than 13 bits of logic 1 can occur, while, at the end of a packet, a sequence of no more than 26 bits of logic 1 can occur.

In another embodiment of the present invention, Arithmetic compression can be used instead of Huffman encoding. The tables for Arithmetic encoding are preferably constructed such that a sequence of 32 bits of logic 1 will not occur in the interior of a message (which is important for a random sign-on in the middle of the stream).

Arithmetic compression provides an advantage of about 6% better compression than Huffman and uses half as much memory for tables, which allows the number of tables to be increased). Indeed, the addition of more tables and/or another level of tables yields more efficient compression. Although Arithmetic compression may take about 6 times as long as Huffman, this can certainly be improved by flattening the subroutine call tree (wherein there is a subroutine call for each output bit.)

In summary, a compression scheme according to one aspect of the invention utilizes a state machine, wherein in each state, there is a compression/decompression table comprising information on how to encode/decode the next character, as well as pointers that indicated which state to go to based on that character. A skeleton of the state machine (nodes and pointers) is preferably built by finding sequences of characters that appear often in the input. Once the skeleton has been determined, a large set of data is run through the system and counts are kept of characters seen in each state. These counts are then used to construct the encode/decode tables for the statistical compression.

Other approaches may be used to build the skeleton of the state machine. A very large fraction of the traffic on a certain feed consists of messages in the digital data feed format, which is fairly constrained. It may be possible to build by hand a skeleton that takes into account this format. For instance, capital letters only appear in the symbol name at the beginning. This long-range context information can be represented with our current approach. Once a basic skeleton is in place, the structure could be extended for sequences that occur frequently.

The above-described statistical compression schemes provide content-dependent compression and decompression. In other words, for a given data stream, the above schemes are preferably structured based on the data model associated with the given stream. It is to be appreciated, however, that other compression schemes may be employed for providing accelerated data transmission in accordance with the present invention for providing effectively increased communication bandwidth and/or reduction in latency. For instance, in another embodiment of the present invention, the data compression/decompression techniques disclosed in the above-incorporated U.S. Pat. No. 6,195,024, entitled "Content Independent Data Compression Method and System" may be used in addition to, or in lieu of, the statistical based compression schemes described above.

In general, a content-independent data compression system is a data compression system that provides an optimal compression ratio for an encoded stream regardless of the data content of the input data stream. A content-independent data compression method generally comprises the steps of compressing an input data stream, which comprises a plurality of disparate data types, using a plurality of different encoders. In other words, each encoder compresses the input data stream and outputs blocks of compressed data. An encoded data stream is then generated by selectively combining compressed data blocks output from the encoders based on compression ratios obtained by the encoders. Because a multitude of different data types may be present within a given input data stream, or data block, to it is often difficult and/or impractical to predict the level of compression that will be achieved by any one encoding technique. Indeed, rather than having to first identify the different data types (e.g., ASCII, image data, multimedia data, signed and unsigned integers, pointers, etc.) comprising an input data stream and selecting a data encoding technique that yields the highest compression ratio for each of the identified data types, content-independent data compression advantageously applies the input data stream to each of a plurality of different encoders to, in effect, generate a plurality of encoded data streams. The plurality of encoders are preferably selected based on their ability to effectively encode different types of input data. Ultimately, the final compressed data stream is generated by selectively combining blocks of the compressed streams output from the plurality of encoders. Thus, the resulting compressed output stream will achieve the greatest possible compression, regardless of the data content.

In accordance with another embodiment of the present invention, a compression system may employ both a content-dependent scheme and a content-independent scheme, such as disclosed in the above-incorporated application Ser. No. 10/016,355. In this embodiment, the content-dependent scheme is used as the primary compression/decompression system and the content-independent scheme is used in place of, or in conjunction with, the content dependent scheme, when periodically checked "compression factor" meets a predetermined threshold. For instance, the compression factor may comprise a compression ratio, wherein the compression scheme will be modified when the compression ratio falls below a certain threshold. Further, the "compression factor" may comprise the latency of data transmission, wherein the data compression scheme with be modified when the latency of data transmission exceeds a predetermined threshold.

Indeed, as explained above, the efficiency of the content-dependent compression/decompression schemes described herein is achieved, e.g., by virtue of the fact that the encoding tables are based on, and specifically designed for, the known data model. However, in situations where the data model is may be modified, the efficiency of the content-dependent scheme may be adversely affected, thereby possibly resulting in a reduction in compression efficiency and/or an increase in the overall latency of data transmission. In such a situation, as a backup system, the data compression controller can switch to a content-independent scheme that provides improved compression efficiency and reduction in latency as compared to the primary content-dependent scheme.

In yet another embodiment of the present invention, when the efficiency of a content-dependent scheme falls below a predetermined threshold based on, e.g., a change in the data structure of the data stream, the present invention preferably

comprises an automatic mechanism to adaptively modify the encoding tables to generate optimal encoding tables (using the process described above with reference to FIG. **3**).

FIG. **5** is a detailed block diagram illustrates an exemplary content-independent data compression system **110** that may be employed herein. Details of this data compression system are provided in U.S. Pat. No. 6,195,024, which is fully incorporated herein by reference. In this embodiment, the data compression system **110** accepts data blocks from an input data stream and stores the input data block in an input buffer or cache **115**. It is to be understood that the system processes the input data stream in data blocks that may range in size from individual bits through complete files or collections of multiple files. Additionally, the input data block size may be fixed or variable. A counter **120** counts or otherwise enumerates the size of input data block in any convenient units including bits, bytes, words, and double words. It should be noted that the input buffer **115** and counter **120** are not required elements of the present invention. The input data buffer **115** may be provided for buffering the input data stream in order to output an uncompressed data stream in the event that, as discussed in further detail below, every encoder fails to achieve a level of compression that exceeds an a priori specified minimum compression ratio threshold.

Data compression is performed by an encoder module **125** that may comprise a set of encoders E**1**, E**2**, E**3** . . . En. The encoder set E**1**, E**2**, E**3** . . . En may include any number "n" (where n may=1) of those lossless encoding techniques currently well known within the art such as run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. It is to be understood that the encoding techniques are selected based upon their ability to effectively encode different types of input data. It is to be appreciated that a full complement of encoders are preferably selected to provide a broad coverage of existing and future data types.

The encoder module **125** successively receives as input each of the buffered input data blocks (or unbuffered input data blocks from the counter module **120**). Data compression is performed by the encoder module **125** wherein each of the encoders E**1** . . . En processes a given input data block and outputs a corresponding set of encoded data blocks. It is to be appreciated that the system affords a user the option to enable/disable any one or more of the encoders E**1** . . . En prior to operation. As is understood by those skilled in the art, such feature allows the user to tailor the operation of the data compression system for specific applications. It is to be further appreciated that the encoding process may be performed either in parallel or sequentially. In particular, the encoders E**1** through En of encoder module **125** may operate in parallel (i.e., simultaneously processing a given input data block by utilizing task multiplexing on a single central processor, via dedicated hardware, by executing on a plurality of processor or dedicated hardware systems, or any combination thereof). In addition, encoders E**1** through En may operate sequentially on a given unbuffered or buffered input data block. This process is intended to eliminate the complexity and additional processing overhead associated with multiplexing concurrent encoding techniques on a single central processor and/or dedicated hardware, set of central processors and/or dedicated hardware, or any achievable combination. It is to be further appreciated that encoders of the identical type may be applied in parallel to enhance encoding speed. For instance, encoder E**1** may comprise two parallel Huffman encoders for parallel processing of an input data block.

A buffer/counter module **130** is operatively connected to the encoder module **125** for buffering and counting the size of each of the encoded data blocks output from encoder module **125**. Specifically, the buffer/counter **130** comprises a plurality of buffer/counters BC**1**, BC**2**, BC**3** . . . BCn, each operatively associated with a corresponding one of the encoders E**1** . . . En. A compression ratio module **135**, operatively connected to the output buffer/counter **130**, determines the compression ratio obtained for each of the enabled encoders E**1** . . . En by taking the ratio of the size of the input data block to the size of the output data block stored in the corresponding buffer/counters BC**1** . . . BCn. In addition, the compression ratio module **135** compares each compression ratio with an a priori-specified compression ratio threshold limit to determine if at least one of the encoded data blocks output from the enabled encoders E**1** . . . En achieves a compression that exceeds an a priori-specified threshold. As is understood by those skilled in the art, the threshold limit may be specified as any value inclusive of data expansion, no data compression or expansion, or any arbitrarily desired compression limit. A description module **138**, operatively coupled to the compression ratio module **135**, appends a corresponding compression type descriptor to each encoded data block which is selected for output so as to indicate the type of compression format of the encoded data block. A data compression type descriptor is defined as any recognizable data token or descriptor that indicates which data encoding technique has been applied to the data. It is to be understood that, since encoders of the identical type may be applied in parallel to enhance encoding speed (as discussed above), the data compression type descriptor identifies the corresponding encoding technique applied to the encoded data block, not necessarily the specific encoder. The encoded data block having the greatest compression ratio along with its corresponding data compression type descriptor is then output for subsequent data processing or transmittal. If there are no encoded data blocks having a compression ratio that exceeds the compression ratio threshold limit, then the original unencoded input data block is selected for output and a null data compression type descriptor is appended thereto. A null data compression type descriptor is defined as any recognizable data token or descriptor that indicates no data encoding has been applied to the input data block. Accordingly, the unencoded input data block with its corresponding null data compression type descriptor is then output for subsequent data processing or transmittal.

Again, it is to be understood that the embodiment of the data compression engine of FIG. **5** is exemplary of a preferred compression system which may be implemented in the present invention, and that other compression systems and methods known to those skilled in the art may be employed for providing accelerated data transmission in accordance with the teachings herein. Indeed, in another embodiment of the compression system disclosed in the above-incorporated U.S. Pat. No. 6,195,024, a timer is included to measure the time elapsed during the encoding process against an a priori-specified time limit. When the time limit expires, only the data output from those encoders (in the encoder module **125**) that have completed the present encoding cycle are compared to determine the encoded data with the highest compression ratio. The time limit ensures that the real-time or pseudo real-time nature of the data encoding is preserved. In addition, the results from each encoder in the encoder module **125** may be buffered to allow additional encoders to be sequentially applied to the output of the previous encoder, yielding a more optimal lossless data compression ratio. Such techniques are discussed in greater detail in the above-incorporated U.S. Pat. No. 6,195,024.

Referring now to FIG. **6**, a detailed block diagram illustrates an exemplary decompression system that may be

employed herein or accelerated data transmission as disclosed in the above-incorporated U.S. Pat. No. 6,195,024. In this embodiment, the data compression engine **180** accepts compressed data blocks received over a communication channel. The decompression system processes the input data stream in data blocks that may range in size from individual bits through complete files or collections of multiple files. Additionally, the input data block size may be fixed or variable.

The data decompression engine **180** comprises an input buffer **155** that receives as input an uncompressed or compressed data stream comprising one or more data blocks. The data blocks may range in size from individual bits through complete files or collections of multiple files. Additionally, the data block size may be fixed or variable. The input data buffer **55** is preferably included (not required) to provide storage of input data for various hardware implementations. A descriptor extraction module **160** receives the buffered (or unbuffered) input data block and then parses, lexically, syntactically, or otherwise analyzes the input data block using methods known by those skilled in the art to extract the data compression type descriptor associated with the data block. The data compression type descriptor may possess values corresponding to null (no encoding applied), a single applied encoding technique, or multiple encoding techniques applied in a specific or random order (in accordance with the data compression system embodiments and methods discussed above).

A decoder module **165** includes one or more decoders D**1** . . . Dn for decoding the input data block using a decoder, set of decoders, or a sequential set of decoders corresponding to the extracted compression type descriptor. The decoders D**1** . . . Dn may include those lossless encoding techniques currently well known within the art, including: run length, Huffman, Lempel-Ziv Dictionary Compression, arithmetic coding, data compaction, and data null suppression. Decoding techniques are selected based upon their ability to effectively decode the various different types of encoded input data generated by the data compression systems described above or originating from any other desired source.

As with the data compression systems discussed in the above-incorporated U.S. Pat. No. 6,195,024, the decoder module **165** may include multiple decoders of the same type applied in parallel so as to reduce the data decoding time. An output data buffer or cache **170** may be included for buffering the decoded data block output from the decoder module **165**. The output buffer **70** then provides data to the output data stream. It is to be appreciated by those skilled in the art that the data compression system **180** may also include an input data counter and output data counter operatively coupled to the input and output, respectively, of the decoder module **165**. In this manner, the compressed and corresponding decompressed data block may be counted to ensure that sufficient decompression is obtained for the input data block.

Again, it is to be understood that the embodiment of the data decompression system **180** of FIG. **6** is exemplary of a preferred decompression system and method which may be implemented in the present invention, and that other data decompression systems and methods known to those skilled in the art may be employed for providing accelerated data transmission in accordance with the teachings herein.

It is to be appreciated that a data transmission acceleration system according to the present invention offers a business model by which market data vendors and users in the financial information services industry can receive various benefits. For example, the present invention affords transparent multiplication of bandwidth with minimal latency. Experiments have shown that increased bandwidth of up to 3 times can be achieved with minimal latency. Furthermore, proprietary hardware, including chip and board designs, as well as custom embedded and application software and algorithms associated with accelerated data transmission provide a cost-effective solution that can be seamlessly integrated with existing products and infrastructure. Moreover, the data acceleration through "real-time" compression and decompression affords a dramatic reduction in ongoing bandwidth costs. Further, the present invention provides mechanism to differentiate data feeds from other vendors via enriched content or quantity of the data feed.

In addition, a data compression scheme according to the present invention provides dramatically more secure and encrypted feed from current levels, thus, providing the ability to employ a secure and accelerated virtual private network over the Internet for authorized subscribers or clients with proprietary hardware and software installed.

Moreover, the present invention offers the ability to reduce a client's ongoing monthly bandwidth costs as an incentive to subscribe to a vendor's data feed service.

The present invention is readily extendable for use on a global computer network such as the Internet. This is significant since it creates a virtual private network and is important for the market data vendors and others due to its reduced cost in closed network/bandwidth solutions. In effect, the data vendors get to "ride for free" over the world's infrastructure, while still providing the same (and enhanced) services to their customers.

In yet another embodiment of the present invention a highly optimized data compression and decompression system is utilized to accelerate data transfers for data transmission feeds. This type of compression achieves very high compression ratios (over 10:1) on financial data feeds such as Nasdaq Quote Dissemination Service Data (NQDS) and SuperMontage Services. The information utilized to develop the methods described herein for Nasdaq has been garnered solely from public knowledge through specifications available from the Nasdaq Trader and Nasdaq websites. The techniques disclosed herein are broadly applicable to all financial data feeds and information or trading services.

Three types of encoding are utilized dependent upon the data fields and packet structure. In the event that a data field is unrecognizable then content independent data compression is preferably used, as previously discussed herein.

Variable Length Encoding

The basic unit of the compression process is the code. Each message field or set of set of fields being compressed together is assigned one or more codes in the range 0 . . . N. The code for a single character field is the ASCII value of the field minus 32 since all characters are in the range 32 to 127.

For various reasons, additional (escape) codes may be added to those for field values. For example, the category field has an escape code to indicate the end of a block and another to allow encoding of messages, which do not match the current format.

A basic technique used is variable rate encoding of symbols. In this approach, different amounts of the output bits are used to transmit the codes within a set. Higher frequency codes use less output bits while lower frequency codes use more output bits. Thus the average number of bits is reduced. Two methods of accomplishing this are used. The faster method uses a variant of Huffman coding while the slower method uses a form of Arithmetic coding.

In Huffman coding, each code is represent by an integral number of bits. The code sizes are computed using the stan-

dard algorithm and then (possibly) adjusted to facilitate table driven decoding (for instance, limiting codes to at most 16 bits). In the table driven decoding method used, there is a 256 element base table and two 256 element forwarding table. At each step, the next 8 bits of the input are used to index into the base table. If the code is represented in no more than 8 bits, it will be found directly. Otherwise, there will be a forwarding entry indicating which forwarding table to use and how many input bits to discard before using the next 8 bits as an index. The entry determining the result also indicates how many bits of the input to discard before processing the next field.

In arithmetic coding, the message is essentially represented as the (approximate) product of fractions with base 16384. The numerators of the fractions are proportional to the frequencies with which the codes appear in the training data. The number of output bits used to represent a code is the base 2 logarithm of the fraction. Thus codes which appear in almost all messages may be represented with fractions of a bit.

Single Character Codes

For arithmetic coding, all single character fields are encoded as the ASCII value−32+the number of escape codes. For Huffman coding, certain single character message fields are encoded in the same way. These include:

MM Trade Desk

Quote Condition

Inside Indicator

Quote Type

Other single character fields, which have a single value that occurs most of the time, are encoded as multiple character fields (see next). In Huffman coding the smallest representation for a code is 1 bit. By combining these fields, we may encode the most common combination of values in 1 bit for the whole set. These include:

Message Category+Message Type

Session Identifier+Originator ID

PMM+Bid Price Denominator+Ask Price Denominator (Quotes)

Inside Status+Inside Type

Inside Bid Denominator+Inside Bid MC

Inside Ask Denominator+Inside Ask MC

UPC Indicator+Short Sale Bid Tick

Market of Origin+Reason

Small Set Multiple Character Codes

Multiple character fields with a small number of common values and certain combinations of single character fields are encoded based on the frequency of the combinations. A list of common combinations is used together with an escape code.

The common combinations are encoded using the corresponding code. All other combinations are encoded by the escape code followed by the (7 bit) ASCII values for the characters in the combination. The fields include the field sets above for Huffman coding as well as the following for both approaches:

Retransmission Requester

MM Location

Currency Code

Large Set Multiple Character Codes

Multiple character alphabetic or alphanumeric fields for which a large number of values are possible (Issue Symbol and MMID/MPID) are encoded as follows. Trailing spaces for Issue Symbols are deleted. Then the result is encoded using:

Variable length codes for a list of the most common values together with escapes for the possible lengths of values not in the list.

A table for the first character of the field.

A table for subsequent characters in the field.

If a value is in the list of most common values, it is encoded with the corresponding code. Otherwise, the value is encoded by sending the escape code corresponding to the (truncated) length of the value, followed by the code for the first character, which is then followed by codes for the remaining characters.

Absolute Numeric Values

Numeric fields are transmitted by sending a variable length code for the number of significant bits of the value followed by the bits of the value other than the most significant bit (which is implicitly 1). For example, 27 (a 5 bit value) would be represented by the code for a 5 bit value followed by the 4 least significant bits (11). These fields include:

Short Bid Price

Long Bid Price

Short Bid Size

Long Bid Size

Short Ask Size

Long Ask Size

Short Inside Bid Size

Long Inside Bid Size

Short Inside Ask Size

Long Inside Ask Size

Relative Numeric Values

Numeric fields expected to be close to the value of numeric values occurring earlier in the message are encoded by encoding the difference between the new value and the base value as follows:

If the difference in non-negative and less than ⅛ of the base value, the difference is encoded by sending a variable length code for the number of significant bits of the difference followed by the bits of the difference other than the most significant bit (which is implicitly 1). Otherwise, the new value is encoded by sending a variable length code for the number of significant bits of the value followed by the bits of the value other than the most significant bit (which is implicitly 1). The difference significant bit codes and the value significant bit codes are mutually exclusive. The following fields are encoded using the difference compared to the field in parentheses:

Short Ask Price (Bid Price)

Long Ask Price (Bid Price)

Short Inside Bid Price (Bid Price)

Short Inside Ask Price (Inside Bid Price)

Long Inside Bid Price (Bid Price)

Long Inside Ask Price (Inside Bid Price)

Differences

Both time and Message Sequence Number are encoded as the difference between the new value and a previous value within the compression block. This is transmitted using a code giving the sign of the difference and the number of significant bits in the absolute value of the difference followed by the bits of the absolute value other than the first.

Date

Each message within a compression block is expected to have the same date. The base date is transmitted at the beginning of the block as 7 bits of year, 4 bits of month and 5 bits of day of the month. If the date of a message is different than

that of the block, a special escape code is used in place of the encoding of the sequence number and time. This is followed by the year, month and day as above followed by the time in seconds (17 bits) and the sequence number (24 bits).

Message Sequence Number and Time

Message time is converted to seconds after midnight. For all retransmitted messages (Retransmission Requester not "O"), the time is transmitted as a 17-bit value followed by the Message Sequence Number transmitted as a 24-bit value. If the date is not the same as the block date, a time value of 0x1ffff is used as an escape code.

For the first original transmission message in a block, the Message Sequence Number and time are transmitted in the same way.

For arithmetic coding of all other original transmission messages in a block, the Message Sequence Number is transmitted as the encoded change from the Message Sequence Number of the preceding original transmission message. Similarly, the time of all other original transmission messages is encoded as the difference from the previous original transmission message. An escape code in the Message Sequence Number Difference Table is used to indicate that the date is not the same as the block date.

Since almost all sequence number changes are 1 and almost all time changes are 0, we can save a bit (while Huffman coding) by encoding time and sequence number together.

This is done as follows: The most common values for both time and sequence number changes are 0 and 1 so there are three possibilities for each: 0, 1 and something else. Together this yields nine possibilities. An escape code is added to indicate a date different from the block date. To transmit the sequence number and time, the code corresponding the correct combination is first sent and then, if the time difference is not 0 or 1, the difference code for time followed by the difference code for sequence number (if required) is sent.

Unexpected Message Types

For administrative messages or non-control messages of unexpected category or type, the body of the message (the part after the header) is encoded as a 10-bit length field followed by the characters of the body encoded as 7-bit ASCII. Any Quotation message with an unexpected Inside Indicator value will have the remainder of the message encoded similarly.

Termination Code and Error Detection

Each compression block is terminated by an escape code of the message header category or category-type table. If this code is not found before the end of the block or if it is found too soon in the block, an error is returned. It is highly unlikely that a transmission error in the compressed packet could result in decoding so as to end at the same place as the original. The exception to this would be errors in transmitting bits values such as date, time or sequence number or the least significant bits of encoded values or changes. For additional error detection, a CRC check for the original could be added to compressed block.

Experimental Results

The aforecited Data Acceleration Methods were successfully applied to data captured on NASDAQ's NQDS feed. The data captured was first analyzed to optimize the Data Acceleration Methods. Essentially two distinct data rates were evaluated; one similar to the upcoming NASDAQ SuperMontage rate of 9.0 Megabits/sec and the second being the maximum data rate of the NQDS feed of 221 Kilobits/sec. In addition, two modes of data acceleration were applied—one utilizing Arithmetic and the other utilizing Huffman techniques.

The Arithmetic routines typically use 40% more CPU time than the Huffman routines and achieve approximately 15% better compression. On average the compression ratio for the SuperMontage data rate (9.0 Megabits/sec) utilizing Arithmetic Mode, yielded a value of 9.528 with a latency under 10.0 ms. This effectively says that the NQDS feed operating at a SuperMontage rate could be transmitted over one T1 line! Further overall latency can be reduced from 500 msec to something approaching 10 milliseconds if routing delays are reduced. Since the amount of data is substantially less, it will be easier and much more cost efficient to reduce routing delays. Further, since the quantity of transmitted bits is substantially smaller, the skew amongst transmitted packets will also be proportionately lower.

The average compression ratio for the standard NQDS data rate (221 Kbits/sec) was 9.3925 for the Arithmetic Mode with a latency under 128 ms. The higher latency is due to the time required to accumulated data for blocking. Since the present invention allows for very high compression ratios with small blocks of data, the latency can be reduced substantially from 128 msec without a loss in compression ratio. This effectively says that the existing NQDS feed could be transmitted over one-half of a 56 Kilobit/sec modem line. Other advantages of using data acceleration according to the invention is that such methods inherently provide (i) a high level of encryption associated with the Arithmetic Mode (with no subsequent impact on latency) and (ii) error detection capability of the decompression methods at the end user site. The first benefit produces additional levels of security for the transmitted data and the second benefit guarantees that corrupted data will not be displayed at the end user site. Furthermore, the need to dynamically compare the redundant data feeds at the end user site is eliminated.

In yet another embodiment of the present invention the aforecited algorithms and all other data compression/decompression algorithms may be utilized in a data field specific compiler that is utilized to create new data feed and data stream specific compression algorithms.

A data field description language is utilized to define a list of possible data fields and parameters along with associated data compression encoders and parameter lists. In one embodiment of the invention the data fields are defined utilizing the following convention:

```
<start list>
<list file name (optional)>
<data field a descriptor, optional parameters>
[data field a compression algorithm x, optional parameters]
<data field b descriptor, optional parameters>
[data field b compression algorithm y, optional parameters]
...
<data field m descriptor, optional parameters>
[data field m compression algorithm n, optional parameters]
<end list>
```

Thus start list and end list are reserved identifiers however any suitable nomenclature can be utilized.

In this simple embodiment of the present invention the list is then submitted to a data compression compiler that accepts the data field list and creates two output files. The first is a data compression algorithm set comprised of data field specific

encoders and the second output file is a data decompression algorithm set comprised of encoded data field specific decoders. In practice this compiler can be implemented in any high level language, machine code, or any variant in between. In addition the language can be Java, r Visual Basic, or another interpreted language to be dynamically operated over the Internet.

More advanced embodiments of the list can be created where the order of the data fields is important to the selection of encoders. In this case the fields are an ordered vector set and the encoders are also an ordered vector set.

```
<start list>
<list file name (optional)>
<ordered data field list 1, optional parameters>
<data field a, optional parameters; data field b, optional parameters;
...; data field n, optional parameters;>
[data field a compression algorithm x, optional parameters; data
field b
    compression algorithm y, optional parameters; ...;data field m
    compression algorithm n]
[data field b compression algorithm x, optional parameters; data
field a
    compression algorithm y, optional parameters; ...;data field m
    compression algorithm n]
<end list>
```

In this more sophisticated embodiment the encoders are selected based upon the data fields and their specific ordering.

In yet another embodiment of the present invention the sets of ordered data fields can be assigned to sets by set name, giving the ability for nesting of sets to facilitate ease of coding.

In yet another embodiment of the present invention the optional parameters to each encoder are utilized to share parameters amongst the same or different data fields.

Although illustrative embodiments have been described herein with reference to the accompanying drawings, it is to be understood that the present invention is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention. All such changes and modifications are intended to be included within the scope of the invention as defined by the appended claims.

What is claimed is:

1. A method of decoding one or more encoded messages of a data packet in a financial data stream using a data decoding engine, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to an encoded message, the method comprising:

receiving an encoded message in a data packet of the financial data stream having a plurality of data fields associated with the encoded message and one or more descriptors comprising one or more values, wherein the one or more descriptors indicate data field types of the data fields and lossless encoders used to encode the data fields, and further wherein the lossless encoders are selected based on analyses of content of the data fields;

analyzing the encoded message to identify a descriptor;

selecting one or more lossless decoders for a data field associated with the encoded message, wherein the selecting is based on the descriptor and a description file, and further wherein the description file comprises data field types and associated lossless decoders;

decoding the data field with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data field is encoded utilizing content dependent data compression; and

decoding the data field with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data field is encoded utilizing content independent data compression.

2. The method of claim **1**, wherein the one or more lossless decoders are further selected based upon the specific ordering of the data field in the encoded message.

3. The method of claim **1**, wherein the descriptor comprises values corresponding to a single applied decompression technique or multiple decompression techniques applied in a specific order.

4. The method of claim **1**, further comprising initiating the method of decoding one or more encoded messages of a data packet in a financial data stream using a synchronization point, wherein the financial data stream includes a plurality of synchronization points.

5. The method of claim **4**, wherein the one or more encoded messages of a data packet are included in a broadcast to a plurality of client systems.

6. The method of claim **1**, wherein decoding the data field comprises packet independent data decoding.

7. The method of claim **1**, further comprising providing one or more global state machines and one or more adaptive local state machines.

8. The method of claim **7**, further comprising:

storing in one or more of the global state machines data fields that are likely to repeat in the financial data stream based on a priori knowledge of the data stream; and

storing in one or more of the adaptive local state machines the decoded data field such that the data field is available to decode one or more other data fields.

9. The method of claim **8**, further comprising resetting one or more of the adaptive local state machines at a determinate point of the data packet.

10. The method of claim **1**, wherein the time of receiving and decoding the one or more encoded messages of a data packet is less than the time to receive the one or more encoded messages of a data packet in unencoded form.

11. The method of claim **1**, wherein the method of decoding one or more encoded messages of a data packet in a financial data stream achieves an expansion ratio of at least 1:10.

12. The method of claim **1**, wherein the method of decoding one or more encoded messages of a data packet in a financial data stream is performed in real-time.

13. A system for encoding a plurality of data blocks to create an encoded data packet in a financial data stream, wherein multiple encoders applying a plurality of lossless compression techniques are applied to a plurality of data blocks, the system comprising:

an input interface that receives a data block from the plurality of data blocks;

a memory with a fixed table of data blocks based on a priori knowledge of the financial data stream and an adaptive table of data blocks;

a data encoding engine operatively connected to said input interface and said memory having a computer readable program code of instructions executable by the data encoding engine, said instructions comprising instructions to:

analyze content of the data block to determine a data block type;

select one or more lossless encoders for the data block based on the data block type and a computer file, wherein the computer file indicates data block types and associated encoders;

encode the data block with a selected lossless encoder using a data block in said adaptive table identified by said selected lossless encoder, if available, otherwise using a data block in said fixed table identified by said selected lossless encoder; and

store the data block in said adaptive table, such that the data block is available to encode one or more other data blocks; and

an output interface operatively connected to said data encoding engine that outputs the encoded data packet with a descriptor comprising one or more values, wherein the descriptor indicates the selected one or more lossless encoders.

14. The system of claim **13**, wherein the data block corresponds to a data field of a message.

15. The system of claim **13**, wherein the memory resets the adaptive table at a determinate point of the data packet.

16. The system of claim **13**, wherein the system for encoding includes a plurality of synchronization points in the financial data stream for initiating decoding the financial data stream.

17. The system of claim **13**, further comprising instructions executable by the data encoding engine to encode one or more data blocks with a selected lossless encoder utilizing content independent data compression, if the data block type is associated with a lossless encoder utilizing content independent data compression.

18. The system of claim **13**, wherein the data encoding engine performs packet independent data encoding.

19. The system of claim **13**, wherein the instructions to encode the data block with a selected lossless encoder comprises using a difference between the data block and a data block in the adaptive table.

20. The system of claim **13**, wherein the system for encoding a plurality of data blocks to create an encoded data packet in a financial data stream achieves a compression ratio of at least 10:1.

21. The system of claim **13**, wherein the system for encoding a plurality of data blocks to create an encoded data packet in a financial data stream operates in real-time.

22. A system for encoding a plurality of data blocks to create an encoded data packet in a financial data stream, wherein multiple encoders applying a plurality of lossless compression techniques are applied to a plurality of data blocks, the system comprising:

an input interface that receives a data block;

a data encoding engine operatively connected to said input interface having a computer readable program code of instructions executable by the data encoding engine, said instructions comprising instructions to:

analyze content of the data block to determine a data block type;

select one or more lossless encoders based on the data block type and a computer file, wherein the computer file indicates data block types and associated encoders;

encode the data block with a selected lossless encoder utilizing content dependent data compression, if the data block type is recognized as associated with a lossless encoder utilizing content dependent data compression; and

encode the data block with a selected lossless encoder utilizing content independent data compression, if the

data block type is not recognized as associated with a lossless encoder utilizing content dependent data compression; and

an output interface operatively connected to said data encoding engine that outputs a descriptor comprising one or more values in the encoded data packet in the financial data stream, wherein the descriptor indicates the one or more selected lossless encoders.

23. The system of claim **22**, wherein the data block corresponds to a data field of a message.

24. The system of claim **22**, wherein the system for encoding includes a plurality of synchronization points in the financial data stream for initiating decoding the financial data stream.

25. The system of claim **22**, wherein the data encoding engine performs packet independent data encoding.

26. The system of claim **22**, further comprising one or more global state machines and one or more adaptive local state machines operatively connected to said data encoding engine.

27. The system of claim **26**, wherein the one or more global state machines store data blocks that are likely to repeat in the financial data stream based on a priori knowledge of the data stream and the one or more adaptive local state machines store the received data block such that the data block is available to encode one or more other data blocks.

28. The system of claim **27**, wherein the one or more adaptive local state machines reset at a determinate point of the data packet.

29. A method of encoding a plurality of data blocks to create an encoded data packet in a financial data stream using a data encoding engine, wherein multiple encoders applying a plurality of lossless compression techniques are applied to a plurality of data blocks, the method comprising:

receiving a data block from the plurality of data blocks;

analyzing content of the data block to determine a data block type;

selecting one or more lossless encoders based on the data block type and a computer file, wherein the computer file indicates data block types and associated encoders;

encoding the data block with a selected lossless encoder utilizing content dependent data compression, if the data block type is recognized as associated with a lossless encoder utilizing content dependent data compression;

encoding the data block with a selected lossless encoder utilizing content independent data compression, if the data block type is not recognized as associated with a lossless encoder utilizing content dependent data compression; and

providing a descriptor for the encoded data packet in the financial data stream, wherein the descriptor indicates the one or more selected lossless encoders for the encoded data block.

30. The method of claim **29**, wherein the data block corresponds to a data field of a message.

31. The method of claim **29**, further comprising including a plurality of synchronization points in the financial data stream for initiating decoding the financial data stream.

32. The method of claim **31**, wherein the encoded data packet in the financial data stream is broadcast to a plurality of client systems.

33. The method of claim **32**, wherein the encoded data packet is a User Datagram Protocol (UDP) data packet.

34. The method of claim **29**, wherein the encoding comprises packet independent data encoding.

35. The method of claim **29**, further comprising providing one or more global state machines and one or more adaptive local state machines.

36. The method of claim **35**, further comprising:

storing in one or more of the global state machines data blocks that are likely to repeat in the financial data stream based on a priori knowledge of the data stream; and

storing in one or more of the adaptive local state machines the received data block such that the data block is available to encode one or more other data blocks.

37. The method of claim **36**, further comprising resetting one or more of the adaptive local state machines at a determinate point of the encoded data packet.

38. The method of claim **29**, wherein encoding the data block utilizing content dependent data compression comprises using a difference between data blocks in the encoded data packet.

39. The method of claim **29**, wherein the time of encoding the plurality of data blocks and transmitting the encoded data packet is less than the time to transmit the plurality of data blocks in unencoded form.

40. The method of claim **29**, wherein the plurality of data blocks includes one or more of stock, options, and futures information.

41. The method of claim **29**, wherein the descriptor comprises values corresponding to a single applied compression technique or multiple compression techniques applied in a specific order.

42. The method of claim **29**, wherein the method of encoding a plurality of data blocks to create an encoded data packet in a financial data stream is performed in real-time.

43. A method of encoding a plurality of data blocks to create an encoded data packet for a financial data stream using a data encoding engine, wherein multiple encoders applying a plurality of lossless compression techniques are applied to a plurality of data blocks, the method comprising:

providing a fixed table of data blocks based on a priori knowledge of the financial data stream;

providing an adaptive table of data blocks;

receiving a data block from the plurality of data blocks;

analyzing content of the data block to determine a data block type;

selecting one or more lossless encoders for the data block based on the data block type and a computer file, wherein the computer file indicates data block types and associated encoders;

encoding the data block with a selected lossless encoder using a data block in said adaptive table identified by said selected lossless encoder, if available, otherwise using a data block in said fixed table identified by said selected lossless encoder;

storing the data block in said adaptive table, such that the data block is available to encode one or more other data blocks; and

providing a descriptor for the encoded data packet, wherein the descriptor indicates the selected one or more lossless encoders for the encoded data block.

44. The method of claim **43**, wherein the data block corresponds to a data field of a message.

45. The method of claim **43**, further comprising resetting the adaptive table at a determinate point of the encoded data packet.

46. The method of claim **43**, further comprising including a plurality of synchronization points in the financial data stream for initiating decoding the financial data stream.

47. The method of claim **43**, wherein encoding the data block comprises packet independent data encoding.

48. The method of claim **43**, further comprising encoding one or more of the plurality of data blocks with a selected lossless encoder utilizing content independent data compression, if the data block type is associated with a lossless encoder utilizing content independent data compression.

49. The method of claim **43**, wherein encoding the data block with a selected lossless encoder comprises using a difference between the data block and a data block in the adaptive table.

50. The method of claim **43**, wherein the method of encoding a plurality of data blocks to create an encoded data packet in a financial data stream achieves a compression ratio of at least 10:1.

51. The method of claim **43**, wherein the method of encoding a plurality of data blocks to create an encoded data packet in a financial data stream is performed in real-time.

52. A method of encoding one or more messages to create an encoded data packet for a financial data stream using a data encoding engine, wherein multiple encoders applying a plurality of lossless compression techniques are applied to a plurality of data fields of a message, the method comprising:

providing a fixed table of data fields based on a priori knowledge of the financial data stream;

providing an adaptive table of data fields;

receiving a message from the one or more messages;

analyzing content of a data field in the message to determine a data field type;

selecting one or more lossless encoders for the data field based on the data field type and a computer file, wherein the computer file indicates data block types and associated encoders;

encoding the data field with a selected lossless encoder using a data field in said adaptive table identified by said selected lossless encoder, if available, otherwise using a data field in said fixed table identified by said selected lossless encoder;

storing the data field in said adaptive table, such that the data field is available to encode one or more other data fields; and

providing a descriptor for the encoded data packet, wherein the descriptor indicates the selected one or more lossless encoders for the encoded data field.

53. The method of claim **52**, further comprising resetting the adaptive table at a determinate point of the data packet.

54. The method of claim **52**, further comprising including a plurality of synchronization points in the financial data stream for initiating decoding the financial data stream.

55. The method of claim **52**, wherein encoding the data field comprises packet independent data encoding.

56. The method of claim **52**, further comprising encoding one or more data fields with a selected lossless encoder utilizing content independent data compression, if the data field type is associated with a lossless encoder utilizing content independent data compression.

57. The method of claim **52**, wherein encoding the data field with a selected lossless encoder comprises using a difference between the data field and a data field in the adaptive table.

58. The method of claim **52**, wherein the method of encoding one or more messages to create an encoded data packet for a financial data stream is performed in real-time.

59. The method of claim **52**, wherein the method of encoding one or more messages to create an encoded data packet for a financial data stream achieves a compression ratio of at least 10:1.

60. A system for encoding one or more messages to create an encoded data packet in a financial data stream, wherein multiple encoders applying a plurality of lossless compres-

sion techniques are applied to a plurality of data fields of a message, the system comprising:

an input interface that receives a message, wherein the message comprises a plurality of data fields;

a data encoding engine operatively connected to said input interface having a computer readable program code of instructions executable by the data encoding engine, said instructions comprising instructions to:

analyze content of a data field of the message to determine a data field type;

select one or more lossless encoders based on the data field type and a description file, wherein the description file indicates data field types and associated encoders;

encode the data field with a selected lossless encoder utilizing content dependent data compression, if the data block type is recognized as associated with a lossless encoder utilizing content dependent data compression; and

encode the data field with a selected lossless encoder utilizing content independent data compression, if the data block type is not recognized as associated with a lossless encoder utilizing content dependent data compression; and

an output interface operatively connected to said data encoding engine that outputs a descriptor in the encoded data packet in the financial data stream, wherein the descriptor indicates the selected one or more lossless encoders.

61. The system of claim **60**, wherein the system for encoding includes a plurality of synchronization points in the financial data stream for initiating decoding the financial data stream.

62. The system of claim **61**, wherein the encoded data packet is included in a broadcast to a plurality of client systems.

63. The system of claim **60**, wherein the data encoding engine performs packet independent data encoding.

64. The system of claim **60**, further comprising one or more global state machines and one or more adaptive local state machines operatively connected to said data encoding engine.

65. The system of claim **64**, wherein the one or more global state machines store data fields that are likely to repeat in the financial data stream based on a priori knowledge of the data stream and the one or more adaptive local state machines store the data field such that the data field is available to encode one or more other data fields.

66. The system of claim **65**, wherein the one or more adaptive local state machines reset at a determinate point of the data packet.

67. The system of claim **60**, wherein the instructions to encode the data field utilizing content dependent data compression comprises using a difference between the content of data fields in the encoded data packet.

68. The system of claim **60**, wherein the time of encoding the one or more messages and transmitting the encoded data packet in the financial data stream is less than the time to transmit the one or more messages in unencoded form.

69. The system of claim **60**, wherein the one or more messages include one or more of stock, options, and futures information.

70. The system of claim **60**, wherein the one or more messages include financial news.

71. The system of claim **60**, wherein the system for encoding one or more messages to create an encoded data packet in a financial data stream achieves a compression ratio of at least 10:1.

72. The system of claim **60**, wherein the system for encoding one or more messages to create an encoded data packet in a financial data stream operates in real-time.

73. A method of decoding one or more encoded data packets of a financial data stream using a data decoding engine, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to an encoded data packet, the method comprising:

receiving an encoded data packet from the financial data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders used to encode data blocks associated with the encoded data packet, and further wherein the lossless encoders are selected based on analyses of content of the data blocks;

analyzing the encoded data packet of the financial data stream to identify a descriptor;

selecting one or more lossless decoders for a data block associated with the data packet, wherein the selecting is based on the descriptor;

decoding the data block with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data block is encoded utilizing content dependent data compression; and

decoding the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression.

74. The method of claim **73**, wherein the data block corresponds to a data field associated with a message in the encoded data packet of the financial data stream.

75. The method of claim **73**, wherein the descriptor comprises values corresponding to a single applied decompression technique or multiple decompression techniques applied in a specific order.

76. The method of claim **73**, wherein decoding the data block utilizing content independent data decompression occurs prior to decoding the data block utilizing content dependent data decompression.

77. The method of claim **73**, further comprising initiating the method of decoding one or more encoded data packets of a financial data stream using a synchronization point, wherein the financial data stream includes a plurality of synchronization points.

78. The method of claim **77**, wherein the one or more encoded data packets of the financial data stream are broadcast to a plurality of client systems.

79. The method of claim **78**, wherein the one or more encoded data packets are User Datagram Protocol (UDP) data packets.

80. The method of claim **73**, wherein decoding the data block comprises packet independent data decoding.

81. The method of claim **73**, further comprising providing one or more global state machines and one or more adaptive local state machines.

82. The method of claim **81**, further comprising:

storing in one or more of the global state machines data blocks that are likely to repeat in the financial data stream based on a priori knowledge of the data stream; and

storing in one or more of the adaptive local state machines the decoded data block such that the data block is available to decode one or more other data blocks.

83. The method of claim **82**, further comprising resetting one or more of the adaptive local state machines at a determinate point of the data packet.

84. The method of claim **73**, wherein decoding the data block utilizing content dependent data decompression comprises using a difference between data blocks in the encoded data packet.

85. The method of claim **73**, wherein the time of receiving and decoding the one or more encoded data packets is less than the time to receive the one or more encoded data packets in unencoded form.

86. The method of claim **73**, wherein the one or more encoded data packets include one or more of stock, options, and futures information.

87. The method of claim **73**, wherein the one or more encoded data packets include financial news.

88. The method of claim **73**, wherein the method of decoding one or more encoded data packets of a financial data stream achieves an expansion ratio of at least 1:10.

89. The method of claim **73**, wherein the method of decoding one or more encoded data packets of a financial data stream is performed in real-time.

90. The method of claim **73**, wherein the one or more encoded data packets of the financial data stream is transmitted to one or more client systems utilizing TCP/IP.

91. A method of decoding one or more encoded data packets in a financial data stream using a data decoding engine, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to an encoded data packet, the method comprising:

providing a fixed table of data blocks based on a priori knowledge of the financial data stream;

providing an adaptive table of data blocks;

receiving an encoded data packet from the financial data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders used to encode data blocks associated with the encoded data packet, and further wherein the lossless encoders are selected based on analyses of content of the data blocks;

analyzing the encoded data packet to identify a descriptor;

selecting one or more lossless decoders for a data block associated with the encoded data packet, wherein the selecting is based on the descriptor;

decoding the data block with a selected lossless decoder using a data block in said adaptive table identified by said selected lossless decoder, if available, otherwise using a data block in said fixed table identified by said selected lossless decoder; and

storing the decoded data block in said adaptive table, such that the decoded data block is available to decode one or more other data blocks.

92. The method of claim **91**, wherein the data block corresponds to a data field associated with a message in the encoded data packet of the financial data stream.

93. The method of claim **91**, further comprising resetting the adaptive table at a determinate point of the encoded data packet.

94. The method of claim **91**, further comprising initiating the method of decoding one or more encoded data packets in a financial data stream using a synchronization point, wherein the financial data stream includes a plurality of synchronization points.

95. The method of claim **91**, wherein decoding the data block comprises packet independent data decoding.

96. The method of claim **91**, further comprising decoding one or more data blocks with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression.

97. The method of claim **91**, wherein decoding the data block with a selected lossless decoder comprises using a difference between the data block and a data block in the adaptive table.

98. A system for decoding one or more encoded data packets of a financial data stream, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to an encoded data packet, the system comprising:

an input interface that receives an encoded data packet from the financial data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders used to encode data blocks associated with the encoded data packet, and further wherein the lossless encoders are selected based on analyses of content of the data blocks;

a data decoding engine operatively connected to said input interface having a computer readable program code of instructions executable by the data decoding engine, said instructions comprising instructions to:

analyze the encoded data packet of the financial data stream to identify a descriptor;

select one or more lossless decoders for a data block associated with the encoded data packet, wherein the selecting is based on the descriptor;

decode the data block with a selected lossless decoder utilizing content dependent data decompression, if the descriptor indicates the data block is encoded utilizing content dependent data compression; and

decode the data block with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression; and

an output interface operatively connected to said data decoding engine that outputs data from the data packet.

99. The system of claim **98**, wherein the data block corresponds to a data field associated with a message in the encoded data packet of the financial data stream.

100. The system of claim **98**, wherein the descriptor comprises values corresponding to a single applied decompression technique or multiple decompression techniques applied in a specific order.

101. The system of claim **98**, wherein the system for decoding initiates decoding the one or more encoded data packets of the financial data stream using a synchronization point, and further wherein the financial data stream includes a plurality of synchronization points.

102. The system of claim **98**, wherein the data decoding engine performs packet independent data decoding.

103. The system of claim **98**, further comprising one or more global state machines and one or more adaptive local state machines operatively connected to said data decoding engine.

104. The system of claim **103**, wherein the one or more global state machines store data blocks that are likely to repeat in the financial data stream based on a priori knowledge of the data stream and the one or more adaptive local state machines store the decoded data block such that the data block is available to decode one or more other data blocks.

105. The system of claim **104**, wherein the one or more adaptive local state machines reset at a determinate point of the encoded data packet.

106. The system of claim **98**, wherein the system for decoding one or more encoded data packets of a financial data stream achieves an expansion ratio of at least 1:10.

107. The system of claim **98**, wherein the system for decoding one or more encoded data packets of a financial data stream operates in real-time.

108. A system for decoding one or more data packets in a financial data stream, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to an encoded data packet, the system comprising:

an input interface that receives an encoded data packet from the financial data stream having one or more descriptors comprising one or more values, wherein the one or more descriptors indicate lossless encoders used to encode data blocks associated with the data packet, and further wherein the lossless encoders are selected based on analyses of content of the data blocks;

a memory with a fixed table of data blocks based on a priori knowledge of the financial data stream and an adaptive table of data blocks;

a data decoding engine operatively connected to said input interface and said memory having a computer readable program code of instructions executable by the data decoding engine, said instructions comprising instructions to:

analyze the encoded data packet to identify a descriptor;

select one or more lossless decoders for a data block associated with the encoded data packet, wherein the selecting is based on the descriptor;

decode the data block with a selected lossless decoder using a data block in said adaptive table identified by said selected lossless decoder, if available, otherwise using a data block in said fixed table identified by said selected lossless decoder; and

store the decoded data block in said adaptive table, such that the decoded data block is available to decode one or more other data blocks; and

an output interface operatively connected to said data decoding engine that outputs data from the data packet.

109. The system of claim **108**, wherein the data block corresponds to a data field associated with a message in the encoded data packet of the financial data stream.

110. The system of claim **108**, wherein the memory resets the adaptive table at a determinate point of the data packet.

111. The system of claim **108**, wherein the system for decoding initiates decoding the one or more data packets in a financial data stream using a synchronization point, and further wherein the financial data stream includes a plurality of synchronization points.

112. The system of claim **108**, wherein the data decoding engine performs packet independent data decoding.

113. The system of claim **108**, further comprising instructions executable by the data decoding engine to decode one or more data blocks with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression.

114. The system of claim **108**, wherein the instructions to decode the data block with a selected lossless decoder comprises using a difference between the data block and a data block in the adaptive table.

115. A system for decoding one or more encoded messages of a data packet in a financial data stream, wherein multiple decoders applying a plurality of lossless decompression techniques are applied to an encoded message, the system comprising:

an input interface that receives an encoded message in a data packet from the financial data stream having a plurality of data fields associated with the encoded message and one or more descriptors comprising one or more values, wherein the one or more descriptors indicate data field types of the data fields and lossless encoders used to encode the data fields, and further wherein the lossless encoders are selected based on analyses of content of the data fields;

a memory with a fixed table of data fields based on a priori knowledge of the financial data stream and an adaptive table of data fields;

a data decoding engine operatively connected to said input interface and said memory having a computer readable program code of instructions executable by the data decoding engine, said instructions comprising instructions to:

analyze the encoded message to identify a descriptor;

select one or more lossless decoders for a data field associated with the encoded message, wherein the selecting is based on the descriptor and a description file, and further wherein the description file comprises data field types and associated lossless decoders;

decode the data field with a selected lossless decoder using a data field in said adaptive table identified by said selected lossless decoder, if available, otherwise using a data field in said fixed table identified by said selected lossless decoder; and

store the decoded data field in said adaptive table, such that the decoded data field is available to decode one or more other data fields; and

an output interface operatively connected to said data decoding engine that outputs data from the data packet.

116. The system of claim **115**, wherein the memory resets the adaptive table at a determinate point of the data packet.

117. The system of claim **115**, wherein the system for decoding initiates decoding the one or more encoded messages using a synchronization point, wherein the financial data stream includes a plurality of synchronization points.

118. The system of claim **115**, wherein the data decoding engine performs packet independent data decoding.

119. The system of claim **115**, wherein the instructions to select one or more lossless decoders is further based upon the specific ordering of the data field in the encoded message.

120. The system of claim **115**, further comprising instructions executable by the data decoding engine to decode the data field with a selected lossless decoder utilizing content independent data decompression, if the descriptor indicates the data block is encoded utilizing content independent data compression.

121. The system of claim **115**, wherein the instructions to decode the data block with a selected lossless decoder comprises using a difference between the data field and a data field in the adaptive table.

122. The system of claim **115**, wherein the system for decoding one or more encoded messages of a data packet in a financial data stream achieves an expansion ratio of at least 1:10.

123. The system of claim **115**, wherein the system for decoding one or more encoded messages of a data packet in a financial data stream operates in real-time.

* * * * *

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

REALTIME DATA, LLC d/b/a IXO, *Plaintiff,* vs. MORGAN STANLEY, ET AL., *Defendants.*	Case No. 1:11-cv-6696-KBF 1:11-cv-6701-KBF 1:11-cv-6704-KBF JURY TRIAL DEMANDED ECF Case
REALTIME DATA, LLC d/b/a IXO, *Plaintiff,* vs. CME GROUP INC., ET AL., *Defendants.*	Case No. 1:11-cv-6697-KBF 1:11-cv-6699-KBF 1:11-cv-6702-KBF JURY TRIAL DEMANDED ECF Case
REALTIME DATA, LLC d/b/a IXO, *Plaintiff,* vs. THOMSON REUTERS, ET AL., *Defendants.*	Case No. 1:11-cv-6698-KBF 1:11-cv-6700-KBF 1:11-cv-6703-KBF JURY TRIAL DEMANDED ECF Case

o ~~[PROPOSED]~~ ORDER ADOPTING THE PARTIES' AGREED CLAIM CONSTRUCTIONS

The Court, having considered the parties' agreed claim constructions, hereby adopts their agreed constructions. The following claim terms and phrases in U.S. Patent Nos. 7,417,568 (the

"'568 patent"), 7,714,747 (the "'747 patent"), and 7,777,651 (the "'651 patent") are construed as set forth below:

Claim Term or Phrase	Patents and Claims	Construction
"analyzing [or analyze] content of the [or a] data block [or field of the message] to determine a data block [or field] type" "analyses of content of [the] data blocks [or fields]"	'747 patent: 1, 8, 14, 19; '651 patent: 1, 13, 22, 29, 43, 60, 91, 108, 115.	directly examining the content of the data to be compressed to determine the data block (*or* data field) type of that data
"analyzing (or analyze) the data packet (or encoded data packet or encoded message) to identify a descriptor"	'651 patent: 1, 13, 22, 29, 43, 60, 91, 108, 115; '747 patent: 1, 8, 14, 19.	examining (*or* examine) the data packet (*or* encoded data packet *or* encoded message) to identify a descriptor in the packet
"recognizing a data field type of a data field"	'568 patent: 1, 20.	recognizing a data field type by analyzing the content of the data field
"packet" "data packet"	'568 patent: 1, 15, 32; '747 patent: 1, 7, 8, 13, 14, 19; '651 patent: 1, 4, 12, 13, 15, 18, 21, 22, 25, 29, 34, 37, 38, 43, 45, 47, 51, 60, 63, 67, 72, 91, 92, 93, 94, 95, 108, 110, 111, 112, 115, 116, 118, 123.	Information limited in type, format, and content and able to be transmitted as a unit across a packet-switched network, the packet including control information that enables the packet to be delivered to an intended destination in the network
"description file"	'568 patent: 20, 22; '651 patent: 1, 60, 115.	a computer file comprising a list created using a data field description language
"data block types and associated lossless encoders"	'747 patent: 14, 19.	Data block types and lossless encoders associated with those data block types

Claim Term or Phrase	Patents and Claims	Construction
"data field types and associated lossless decoders"	'651 patent: 1, 115.	Data field types and lossless decoders associated with those data field types
"data block types and associated encoders"	'651 patent: 13, 22, 29, 43.	Data block types and encoders associated with those data block types
"data field types and associated encoders"	'651 patent: 60.	Data field types and encoders associated with those data field types
"computer file"	'651 patent: 13, 22, 29, 43; '747 patent: 14, 19.	A collection of related records treated as a unit within a computer
"a priori knowledge of the financial data stream"	'651 patent: 13, 43, 91, 108, 115.	prior knowledge of at least some aspects of the content or structure of the financial data stream
"adaptive local state machine"	'651 patent: 7, 26, 35, 37, 64.	hardware, microinstruction code, or application program whose state transitions can change during compression (or decompression) and are based on data values in a data stream
"adaptive table"	'651 patent: 13, 15, 19, 43, 45, 49, 91, 93, 97, 108, 110, 114, 115, 116, 121.	a modifiable table whose data can be changed during compression (or decompression) based on the content of the data stream
"fixed table"	'651 patent: 13, 43, 91, 108, 115.	a predefined, constant table that is used during compression (or decompression)
"the encoding does not require packet-to-packet data dependency"	'568 patent: 32.	The encoding of a data packet is independent of any other data packet in the data stream
"packet independent data encoding"	'651 patent: 18, 25, 34, 47, 63.	Data encoding in which each packet is encoded independent of any other data packet in the data stream
"packet independent data compression"	'568 patent: 15.	Data compression in which each packet is compressed independent of any other data packet in the data stream
"packet independent data decoding"	'651 patent: 95, 112, 118.	Data decoding in which each packet is decoded independent of any other data packet in the data stream

Claim Term or Phrase	Patents and Claims	Construction
"synchronization point[s]"	'651 patent: 4, 16, 24, 46, 61, 94, 111, 117.	an identifiable sequence of one or more bytes in the data stream
"compress" "compressed" "compressing" "compression"	'568 patent: 1, 15, 20, 22, 23; '747 patent: 1, 7, 8, 13, 14, 18, 19, 22; '651 patent: 1, 13, 22, 29, 38, 43, 60, 67.	Compress[ed, ing]: represent [*or* represented *or* representing] data with fewer bits Compression: representation of data with fewer bits
"decompress" "decompressed" "decompressing" "decompression"	'747 patent: 1, 7, 8, 13; '651 patent: 1, 91, 108, 115.	decompress[ed, ing]: represent [*or* represented *or* representing] data with more bits decompression: representation of data with more bits
"wherein the lossless encoders are selected based on analyses of content of the data blocks [or fields]"	'747 patent: 1, 8; '651 patent: 1, 91, 108, 115.	the system (or method) selects the lossless encoders based on analyses of content of the data blocks (or data fields)
"global state machine"	'651 patent: 7, 26, 35, 64.	hardware, microinstruction code, or application program whose state transitions are based on prior knowledge of at least the data stream packet structure
"a determinate point of the [encoded] data packet"	'651 patent: 15, 37, 45, 93, 110, 116.	an identifiable sequence of one or more bytes in the data stream

To the extent an identified claim term in the foregoing list uses language that is construed by the Court in connection with a another claim term, the construction adopted by the Court should be incorporated into the construction in this list.

SO ORDERED:

Dated: New York, New York
July 9, 2012

K B. Forrest
KATHERINE B. FORREST
United States District Judge

CERTIFICATE OF SERVICE

I, Dominick Vitaliano, hereby certify that on the 6th day of March 2013, I caused one copy of the documents listed below:

NON-CONFIDENTIAL BRIEF FOR PLAINTIFF-APPELLANT REALTIME DATA, LLC (doing business as IXO)

to be filed by CM/ECF with:

Clerk of Court
United States Court of Appeals for the Federal Circuit
717 Madison Place, N.W.
Washington, D.C. 20439
Tel: (202) 275-8000
Fax: (202) 275-9678

and to be served via CM/ECF on each of the following:

Morgan Stanley, The Goldman Sachs Group, Inc, J.P. Morgan Chase & Co., Morgan Stanley & Co., Inc., Goldman Sachs & Co., Goldman Sachs Execution & Clearing, LP, J.P. Morgan Securities, Inc., and J. P. Morgan Clearing Corp. (formerly known as Bear, Stearns Securities, Corp.):

Daniel Alexander Devito
Skadden, Arps, Slate, Meagher & Flom
LLP (DC)
Four Times Square
New York, NY 10036
212-735-3210
Fax: 917-777-3210
Email: daniel.devito@skadden.com

Credit Suisse Holdings (USA), Inc. and Credit Suisse Securities (USA) LLC:

Gregory Hayes Lantier
Wilmer Cutler Pickering Hale & Dorr
L.L.P. (DC)
1875 Pennsylvania Avenue, NW
Washington, DC 20006
(202) 663-6000
Fax: (202) 663-6363
Email: gregory.lantier@wilmerhale.com

William F. Lee
Wilmer Cutler Pickering Hale and Dorr LLP
60 State Street
Boston, MA 02109
(617) 526-6000
Email: William.Lee@wilmerhale.com

HSBC Bank USA, N.A. and HSBC Securities (USA) Inc.:

Roy W. Hardin
Locke, Lord, Bissell & Liddell, LLP
2200 Ross Avenue
Suite 2200
Dallas, TX 75201
214-740-8000
Email: rhardin@lockelord.com

Michael Scott Fuller
Locke Lord Bissell & Liddell LLP - Dallas
2200 Ross Ave
Suite 2200
Dallas, TX 75201
214-740-8601
Fax: 214-756-8601
Email: sfuller@lockelord.com

CME Group Inc., Board of Trade of The City of Chicago, Inc., and New York Mercantile Exchange, Inc. (agent of Nymex):

Scott F Partridge
Baker Botts LLP
910 Louisiana
Suite 3000 One Shell Plaza
Houston, TX 77002-4995
713/229-1569
Fax: 713/229-7769
Email: scott.partridge@bakerbotts.com

BATS Trading, Inc. (also known as BATS Exchange, Inc):

John M Challis
Polsinelli Shughart PC
100 S. 4th Street, Suite 1000
St. Louis, MO 63102
(314) 889-7057
Fax: (314) 622-6757
Email: jchallis@polsinelli.com

International Securities Exchange:

Michael Martin Murray
Winston & Strawn LLP (NY)
200 Park Avenue
New York, NY 10166
(212) 294-3510
Fax: (212) 294-4700
Email: mmurray@winston.com

NASDAQ OMX Group, Inc and NASDAQ OMX PHLX, Inc.:

Lynn Eileen Rzonca
Ballard Spahr LLP
1735 Market Street
51st Floor
Philadelphia, PA 19103-7599
(215) 665-8500

Fax: (215) 864-8999
Email: rzoncal@ballardspahr.com

NYSE Euronext, Options Price Reporting Authority, NYSE Arca, Inc., NYSE MKT, LLC (formerly known as NYSE Amex, LLC), and Securities Industry Automation Corporation:

James Holman Shalek
Proskauer Rose LLP
11 Times Square
New York, NY 10036
212-969-3050
Fax: 212-969-2900
Email: jshalek@proskauer.com

Thomson Reuters Corporation:

Constance Sue Huttner
Vinson & Elkins L.L.P.
666 Fifth Avenue, 26th Floor
New York 10103
(212)-237-0040
Fax: 917-849-5339
Email: chuttner@velaw.com

Bloomberg L.P.:

John M DiMatteo
Willkie Farr & Gallagher LLP (NY)
787 Seventh Avenue
New York, NY 10019
(212) 728-8000
Fax: (212) 728-8111
Email: maosdny@willkie.com

FactSet Research Systems Inc.:

Brian E Moran
Robinson & Cole LLP - Stamford
1055 Washington Blvd
Stamford, CT 06901-2249
203/462-7512
Fax: 203.462-7599
Email: bmoran@rc.com

Interactive Data Corporation:

Benjamin W Hattenbach
Irell & Manella LLP
1800 Avenue of the Stars
Suite 900
Los Angeles, CA 90067-4276
310/277-1010
Fax: 310-203-7199
Email: bhattenbach@irell.com

I declare that I am employed by McKool Smith P.C. at whose direction the service was made. Executed on March 6, 2013, at New York, NY.

<u>/s/ Dominick Vitaliano</u>
Dominick Vitaliano
Senior Paralegal
McKool Smith, P.C.
One Bryant Park, 47th floor
New York, New York 10036
212.402.9439 (t)
212.402.9444 (f)
dvitaliano@mckoolsmith.com

CERTIFICATE OF COMPLIANCE

I certify that the foregoing NON-CONFIDENTIAL BRIEF FOR PLAINTIFF-APPELLANT REALTIME DATA, LLC (doing business as IXO):

1. complies with the type-volume limitation of Fed. R. App. P. 32(a)(7)(B). This brief contains 13,993 words, excluding the parts of the brief exempted by Fed. R. App. P. 32(a)(7)(B)(iii) and Fed. Cir. R. 32(b). Microsoft Word 2003 was used to calculate the word count; and

2. complies with the typeface requirements of Fed. R. App. P. 32(a)(5) and the type style requirements of Fed. R. App. P. 32(a)(6). This brief has been prepared in proportionally-spaced typeface using Microsoft Word 2003 in 14-point Times New Roman type style.

Dated: March 6, 2013

Respectfully submitted,

/s/ Dirk D. Thomas

Dirk D. Thomas
MCKOOL SMITH P.C.
1999 K Street, Suite 600
Washington, DC 20006
(202) 370-8302

Attorney for Plaintiff-Appellant Realtime Data, LLC (doing business as IXO)